Less Transmissions, More Throughput: Bringing Carpool to Public WLANs

Wei Wang, *Student Member, IEEE,* Yinejie Chen, Qian Zhang, *Fellow, IEEE*
Kaishun Wu, *Member, IEEE,* and Jin Zhang, *Member, IEEE,*.

Abstract—A typical scenario for public WLANs is large audience environment where Wi-Fi hotspots serve scores of mobile devices. The performance of those Wi-Fi hotspots is extremely poor in terms of low goodput and severe delay due to heavy contention and MAC inefficiency. After carefully investigating the traffic patterns in public WLANs, we propose *Carpool*, a practical design that facilitates transmission sharing among multiple receivers, to tackle this problem. The key idea is to reduce contention by feeding frames for multiple destinations into one transmission at physical layer (PHY). As such, each downlink transmission carries payloads for multiple receivers, which reduces contention overhead and enables in-time response to concurrent requests from multiple users. To achieve efficient and reliable transmission in Carpool, we propose i) a lightweight frame structure to support multiple receivers, and ii) a real-time channel estimation scheme to continuously calibrate channel estimation during the transmission of a Carpool frame. We have implemented the entire PHY of Carpool on the GNURadio/USRP platform and tested it in various indoor environments. Furthermore, our trace-driven MAC evaluation shows that Carpool achieves up to $3.2\times$ goodput gain and reduces up to 75% delay compared to the IEEE 802.11n MAC frame aggregation scheme.

Index Terms—Frame aggregation, phase offset side channel, media access control (MAC), cross-layer, contention reduction

1 Introduction

Over the last decade, the deployments of Wi-Fi-based wireless LANs (WLANs) have expanded rapidly. The popularity of Wi-Fi is largely attributable to the proliferation of Wi-Fi hotspots in public places such as coffee shops, airports and large conventions, where there are a significant number of people accessing the Internet via these Wi-Fi hotspots [1], [2]. These public places are referred to as *large audience environments*. Wi-Fi access points (APs) in large audience environments normally serve a crowd of mobile stations (STAs). Unfortunately, the performance of Wi-Fi hotspots in such crowded environments is extremely poor [3].

The fundamental causes of this problem are high media access control (MAC) overhead and traffic asymmetry. In large audience environments, a significant number of STAs transmit and receive packets in a limited number of channels within a carrier sensing range, which incurs intensive contention. High contention in CSMA-based Wi-Fi MAC leads to substantial overhead, including carrier sensing, backoff, and high collision probability. Moreover, several studies on SIGCOMM traces [4], [5] report that downlink traffic volume is about four times larger than uplink traffic volume. As distributed coordination function (DCF) based Wi-Fi provides equal opportunity for APs and STAs to access channel, this asymmetric traffic pattern results in congested APs and severe downlink throughput degradation [3].

A major solution for this large audience environment scenario is from the view point of coordination and scheduling. Enterprise networks [6], [7] adopt centralized coordination to reduce unnecessary contentions among APs. WiFox [3] prioritizes AP's channel access over STAs to address the traffic asymmetric issue. However, these works focus on coordination and scheduling, which does not address the MAC inefficiency issue in large audience environments. As the most recent standards – IEEE 802.11n and 802.11ac – have largely improved maximum physical layer (PHY) data rates over the previous 802.11a/g from 54 Mbit/s to 600 Mbit/s and over 1Gbit/s, MAC efficiency of Wi-Fi networks degrades rapidly in current high speed Wi-Fi networks due to reduced transmission time for payload. Thus, it is imperative to cope with the inefficiency issue in WLANs.

Instead of considering the traffic asymmetry and the MAC overhead separately, we believe these two issues are correlated and should be considered together. We have the observation that if AP can send data for multiple STAs concurrently in a single transmission, the number of contentions will be reduced, and each downlink transmission can convey more traffic. Although the multi-user frame aggregation can be achieved at MAC layer [8], [9], it suffers from the following limitations in large audience environments. First, there are a bundle of active STAs associated with one AP in large audience environments. Explicitly indicating each receiver's MAC address at header would incur substantial overhead, which compromises the transmission efficiency. Second, the aggregated frame should be reliable when the frame size is very large. In the aggregation

- *W. Wang is with the School of Electronic Information and Communications, Huazhong University of Science and Technology and the Guangzhou Fok Ying Tung Research Institute, Hong Kong University of Science and Technology.*
  Correspondence E-mail: gswwang@cse.ust.hk.
- *Y. Chen and Q. Zhang are with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong.*
- *K. Wu is with the College of Computer Science and Software Engineering, Shenzhen University.*
- *J. Zhang is with the Department of Electrical and Electronic Engineering, South University of Science and Technology of China.*

schemes specified by IEEE 802.11, the maximum length of an aggregated frame can be 64KB, which requires 9.7ms of airtime for transmission at 54Mbit/s. As we allow aggregation across multiple STAs, the frame sizes are larger than those in the IEEE 802.11n aggregation. On the other hand, the coherence time ranges from tens of microseconds to hundreds of milliseconds [10]. Thus, it is very likely that the channel varies during the transmission of a long frame, which hinders the reliability of frame transmission as the tail part of the frame may not be decodable using the outdated channel information.

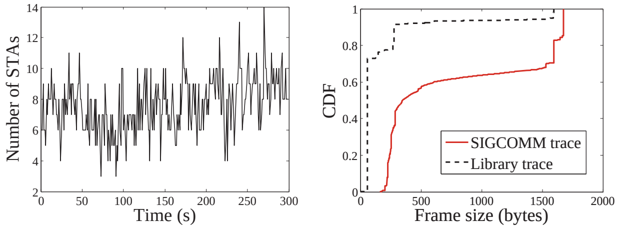
To cope with the aforementioned predicaments, we present *Carpool*, a practical design that enables frame aggregation for multiple receivers at PHY in the orthogonal frequency-division multiplexing (OFDM) based WLANs. Carpool employs a lightweight frame structure to aggregate frames for multiple STAs, and a real-time channel estimation scheme that continuously calibrates the channel estimation in the decoding process. In particular, a Bloom filter [11] based aggregation header is introduced in the new frame format to indicate the receiver of each sub-frame. The aggregation header is placed right after the preamble so that irrelevant STAs within the transmission range can drop the frame without decoding it. To cope with channel variance and guarantee reliable transmission of Carpool frame, we consider correctly decoded data as known "pilots" and use such data pilots in different positions to capture the channel variance in the transmission. To empower real-time channel estimation in existing OFDM PHY, we create a phase offset side channel to check whether an OFDM symbol is correctly decoded, and use the correctly decoded symbols to capture the channel variance during the transmission. Different from existing dirty coding side channels, the phase offset side channel carries a few free bits per symbol by exploiting the inherent redundancy and phase tracking ability of the OFDM-based PHY, while having no impact on data decoding in IEEE 802.11 standards.

The main contributions of this paper are summarized as follows. We analyze the poor performance of Wi-Fi in large audience environments, and propose Carpool to improve efficiency in public WLANs. Specifically, we design a lightweight aggregation mechanism and a real-time channel estimation algorithm based on IEEE 802.11 OFDM PHY to facilitate scalable and reliable transmissions in large audience environments. Furthermore, we implement Carpool on the GNURadio/USRP testbed to demonstrate its feasibility and merits.

The reminder of this paper is structured as follows. We begin in Section 2 with measurements on the traffic patterns in public WLANs. Section 3 presents the overview of Carpool. The aggregation mechanism and real-time channel estimation are elaborated in Section 4 and Section 5, respectively. System implementation of Carpool is described in Section 6, followed by performance evaluation in Section 7. Section 8 discusses several practical considerations of Carpool. Section 9 reviews related work. Finally, Section 10 concludes the paper.

2 Characterizing Traffic in Public WLANs

In this section, we first characterize the features of public WLAN traffic. To further validate the commonly cited characteristics for large audience environments, we measured the downlink traffic in a typical campus library Wi-Fi network. This library measurement is complementary to the SIGCOMM public traces [4], [5] in the following three aspects: i) Library is another large audience environment besides conference; ii) The library measurement captures traces for all STAs in certain locations, while the commonly used SIGCOMM traces [4], [5] are collected from the measurement AP, which associates only with a subset of STAs in the conference; iii) The library WLAN adopts the IEEE 802.11n standard, while the traced SIGCOMM WLANs adopt the IEEE 802.11a/b/g [4], [5].



(a) Concurrent downlink requests. The average number of active STAs per AP is 7.63.

(b) Frame size distribution.

	SIGCOMM'04	SIGCOMM'08	Library
Downlink ratio	80%	83.4%	89.2%

(c) Ratio of downlink traffic volume in public WLANs.

Fig. 1. Traffic statistics in public WLANs.

We setup four sniffers at different locations of the library and capture all traffic on one channel in the campus WLAN. The measurements are conducted during work hours of three working days. The traces include downlink traffic from 15 APs. There are about 164 active STAs on average in five-minute duration, and the number of STAs associated with each AP ranges from 6 to 28. We have the following observations on both SIGCOMM and library traces:

1. Concurrent downlink requests. Fig. 1(a) presents the number of active STAs associated with one AP measured each second using the library trace. Similar measurements on SIGCOMM'08 trace are conducted in [3]. The results show that there are concurrent downlink requests for different STAs in large audience environments.

2. Downlink traffic dominance. Fig. 1(c) shows the ratio of downlink traffic volume. The downlink traffic volume is about four times larger than uplink traffic volume. As distributed coordination function (DCF) based Wi-Fi provides equal opportunity for APs and STAs to access channel, this downlink-uplink asymmetric traffic pattern results in congested APs and severe downlink throughput degradation [3].

3. High ratio of short frames. Fig. 1(b) shows the cumulative distribution function (CDF) of frame sizes in the SIGCOMM and library traces. More than 50% and 90% of the downlink frames are smaller than 300B in the SIGCOMM and library traces, respectively. High ratio of short frames combined with downlink dominance indicates intensive contention in the downlink transmission.

Based on the above features of WLANs traffic, we observe

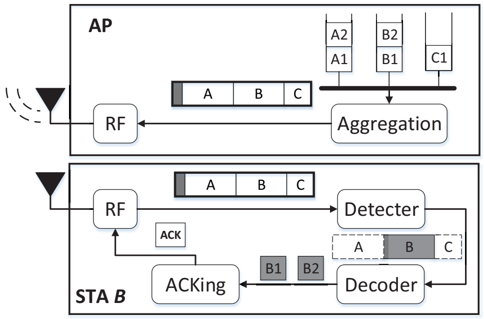


Fig. 2. Architecture of Carpool. The AP aggregates queued frames for different STAs into a single Carpool frame. The STA extracts its frames from the Carpool frame to feed them into the decoder, and feeds back ACK to the AP.

that, it is very inefficient to send one short frame to one STA in each channel access when there are a crowd of STAs waiting for the AP's responses. The contradiction between single-destination transmission and multiple STAs' requests renders downlink traffic backlogged at the AP, and thereby results in low throughput and high delays in the downlink [3].

This contradiction in public WLANs motivates the design of Carpool. Instead of restricting each downlink transmission to one STA, if we can "carpool" frames for multiple STAs into a single transmission, downlink contention would be significantly reduced and each downlink transmission could carry more traffic. One direct benefit of Carpool is the MAC efficiency improvement by reducing the contention overhead. Another benefit of Carpool is that it inherently relieves downlink traffic congestion at AP caused by downlink-uplink traffic asymmetry since each downlink transmission conveys more traffic for multiple receivers. In the following section, we present the detailed PHY/MAC design of Carpool.

3 Overview of Carpool

Carpool is a PHY/MAC design that enables frame aggregation for multiple receivers in OFDM-based WLANs, especially for large audience environments such as conference rooms, airports, and coffee shops. Frames for multiple STAs queued at the Carpool AP are aggregated as a single large frame, where an aggregation header is inserted to indicate the destination of each subframe. Each subframe can be a single frame or multiple frames aggregated at MAC layer that contains the MAC data for one destination. The STAs who hear the frame first detect whether the frame contains payload for them by checking the aggregation header. If an STA detects no subframe for it, the STA drops the whole frame without decoding the payload, otherwise the STA locates its subframe in the frame and only decodes this subframe. Fig. 2 depicts a typical flow of a Carpool frame transmission. The AP aggregates five frames for three STAs into one Carpool frame. STA B checks the header in the Carpool frame and detects its payload in the second subframe. STA B filters out other irrelevant subframes before feeding the frame into the detector. Then, STA B returns an acknowledgement frame (ACK) to the AP. To avoid ACK collisions at AP, receivers need to return ACKs one by one, which can be achieved by simply modifying the Network Allocation Vector (NAV).

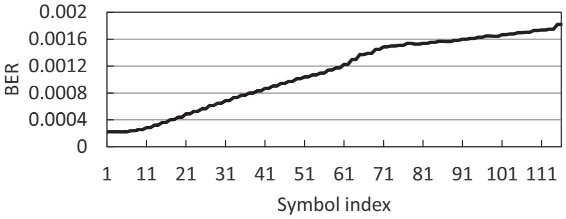


Fig. 3. BER bias in a long frame.

Although the idea seems simple at first glance, it is actually radically challenging to realize Carpool. Simple extensions of current single-destination frame aggregation schemes [9], [12] or existing multi-user aggregation [8], [13] fall short in the following two aspects.

Efficient and scalable aggregation. Compared to traditional frame aggregation, Carpool requires indications of each subframe's destination. A simple solution for this requirement is to add each receiver's MAC address along with the start position of the receiver's frame to the PHY header. However, as PHY header is transmitted at the lowest rate, it would introduce large overhead especially when there are many receivers. We illustrate the overhead of this solution using an example, where 1500B payload for 8 receivers are aggregated into one Carpool frame, and is transmitted at 600Mbit/s. Note that PHY header is transmitted at basic rate of 6.5Mbit/s. As such, the extra time used for transmitting MAC addresses would be $\frac{48\times 8bit}{6.5Mbit/s}=59\mu s$, which is almost three times of the payload transmission time ($\frac{1500\times 8bit}{600Mbit/s}=20\mu s$). Such overhead reduces the aggregation efficiency, and thus is not scalable to many receivers in large audience environments.

Reliable transmission. To aggregate frames for multiple STAs, Carpool has to transmit very large frames. Normally, the size of an IP packet is at most 1500 bytes. The airtime for 1500-byte packet is at most $222\mu s$ at 54Mbit/s. In the aggregation schemes specified by IEEE 802.11n, the maximum length of an aggregated frame can be 64KB, which requires 9.7ms of airtime for transmission at 54Mbit/s. As Carpool allows aggregation across multiple STAs, the frame sizes are larger than those in the IEEE 802.11n aggregation. On the other hand, the coherence time ranges from tens of microseconds to hundreds of milliseconds [10]. Thus, it is very likely that channel varies during the transmission of a long frame. However, in IEEE 802.11, the channel estimation is conducted at the preamble, which only reflects the channel condition at the beginning of the frame transmission, while the channel conditions at the tail part of the frame may be different from the estimated value. As such, the transmission of a long frame is not reliable as the tail part of the frame may not be decodable using the outdated channel estimation.

To validate the above observation on channel variance, we conduct experiments in a 10m × 10m office. We use two USRP nodes as a transmission pair, whose distance is

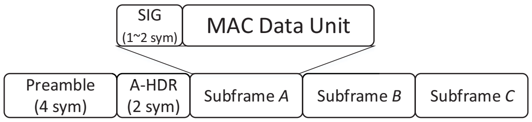


Fig. 4. Frame structure in Carpool.

fixed at 3m. We conduct measurements 10 times in different days and each test lasts 1000 transmissions of 4KB QAM64-modulated frames. Fig. 3 depicts the bit error rate (BER) of different symbols. The symbol index refers to the order of a symbol in the payload, i.e., the position index counts from the first symbol in the payload which is the symbol right after the PHY header. The results show that the BER per symbol increases with the symbol index, meaning that the latter symbols in a long frame are more error-prone. This phenomenon is referred to as *BER bias*. BER bias indicates that the transmission of a long frame is less reliable than that of a short frame. Without addressing this issue, Carpool would incur high retransmission ratio, which compromises the performance gain of aggregation.

The above challenges motivates the careful designs of aggregation mechanism and channel estimation to support Carpool in existing OFDM based WLANs. In Section 4, we present an aggregation mechanism to efficient support Carpool with lightweight overhead and effective ACK feedback. In Section 5, we describe a real-time channel estimation scheme to continuously update the channel conditions during the transmission.

4 Aggregation Mechanism Design

We design an aggregation header in Carpool frame to support multiple receiver aggregation. In the new frame format, a Bloom filter-assisted aggregation header is introduced. We carefully choose a Bloom filter structure to indicate the receivers of the frame and the order of sub-frames. The Bloom filter is placed at the PHY header so that irrelevant STAs within the transmission range can drop the frame without decoding it.

4.1 Frame Aggregation at PHY

Frame structure. Fig. 4 presents the Carpool frame structure. Compared with legacy frame structure, we add an aggregation header (A-HDR) after the preamble. A-HDR consists of two symbols, which are coded using the lowest coding rate, to indicate the receiver of each subframe. A-HDR is followed by a sequence of subframes. Each subframe contains the SIG symbols and MAC data for exactly one receiver. IEEE 802.11 uses one or two symbols for SIG in different modes. The SIG symbols contain information about modulation and coding scheme (MCS) and frame length. Note that the MAC data can be either single data unit or aggregation data unit determined in IEEE 802.11 MAC aggregation (MSDU or MPDU aggregation). Different subframes can adopt different MCSs.

Based on the above frame structure, Carpool receivers detect their subframes as follows. Each receiver hearing the frame first checks A-HDR to find out its intended subframe. Then, for every subframe whose position in the frame is prior to the receiver's subframe, the receiver only decodes the SIG symbol to obtain the subframe's length and then skip the whole subframe. This is feasible in existing IEEE 802.11 PHY as the SIG is not scrambled. After decoding its subframe, the receiver drops all rear subframes. For example, as illustrated in Fig. 4, STA B checks A-HDR and learns that the second subframe contains its payload. Then, STA B decodes the SIG symbols of the first subframe to obtain its length, based on which the location of B's subframe can be computed. STA B only decodes the second subframe and drops the rest of the frame.

A-HDR Design. In A-HDR, we leverage coded Bloom filter [14], [15] to efficiently indicate the receiver of each subframe. The two-symbol A-HDR is coded using BPSK at $1/2$ coding rate, and thereby can be considered as a 48-bit Bloom filter. The Bloom filter is computed by inserting the hash result of each receiver's MAC address one by one. The position information of each subframe is encoded into the selection of hash functions. In particular, we assign a set of h hash functions, referred to as a *hash set*, to each subframe. The ith subframe is assigned with the ith hash set. For the receiver of the ith subframe, we use the ith hash set to hash the receiver's MAC address. Each hash function in the hash set maps the receiver's MAC address to a position in the 48-bit binary vector. As there are h hash functions in a hash set, a MAC address is hashed h times to obtain h positions in the 48-bit binary vector. To insert a MAC address into the Bloom filter, we set the mapped h positions to "1"s. A-HDR is computed by inserting the MAC addresses of all receivers one by one. Fig. 5 gives a toy example to compute the Bloom filter based on two MAC addresses. In this example, A's MAC address is hashed using hash set $\mathbf{H_1}$ as A's payload is in the first subframe. Two hash functions in $\mathbf{H_1}$ map A's MAC address to two positions in a binary vector, and the bits in the corresponding positions in the hash result vector are set to "1"s. B's MAC address is mapped in the same process as A, and the Bloom filter result is computed by combining two hash result vectors using bit-wise "or".

Each receiver checks A-HDR to derive the position of its subframe. A receiver tries each hash set one by one to hash its MAC address into h positions, and checks the positions in the Bloom filter. If any of the bits at the positions computed according to the ith hash set is "0", then the ith subframe is definitely not for the receiver. If all of the bits at the positions are "1"s, the corresponding subframe is considered as a *matched* subframe. As such, the receiver finds and decodes all matched subframes. Note that there are false positives in checking the Bloom filer, that is, the receiver's subframe is a matched subframe while not all matched subframes are the receiver's subframes.

Decoding with false positives. Thanks to the property of Bloom filter, ambiguity in A-HDR will not result in frame loss. As there is no false negative but only false positives, matched subframes may not be the receiver's subframe, but mismatched subframes are definitely not the receiver's subframe. To avoid missing the right subframe, each receiver

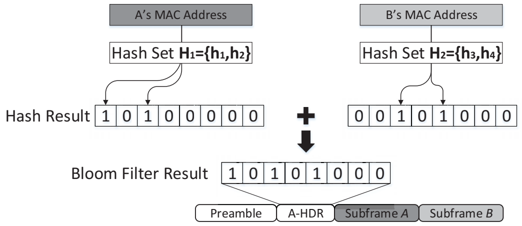


Fig. 5. Illustration of bloom filter

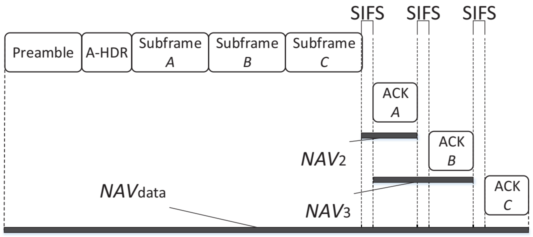


Fig. 6. Illustration of sequential ACK in Carpool.

decodes all matched subframes corresponding to possible indices resolved in the Bloom filter result. To minimize false positive ratio, we carefully determine the optimal number of hashes h in a hash set. We assume that a hash function selects each bit position with equal probability. If there are N receivers in total, for any given hash set, the false positive ratio $r_{\rm FP} = (1-(1-1/48)^{hN})^h \approx (1-e^{\frac{hN}{48}})^h$. The minimal value of $r_{\rm FP}$ is obtained when the first order derivative of $r_{\rm FP}$ equals to zero, where we derive $h = \frac{48}{N}\ln 2$. If the number of receivers is 4-8, the false positive ratio ranges from 0.31% to 5.59%. False positives of 5.59% will not result in frame loss, but only decoding irrelevant frames, which incurs extra power consumption. Detailed power consumption is analyzed in Section 8. In our implementation, we limit the number of receivers to 8, and thus set $h = 4$. Compared to directly adding MAC addresses of all receivers at header, which requires $48 \times 8 = 384$ bits for 8 receivers, A-HDR only has 12.5% overhead.

4.2 Sequential ACK

The downlink aggregation for multiple receivers call for a new design of ACK mechanism. As the differences in the signal propagation delays and processing delays can be very small [16], multiple receivers receive and decode a frame at roughly the same time. In the traditional ACK mechanism, multiple receivers wait for an SIFS interval and return ACKs simultaneously, which would result in collision at AP.

To avoid ACK collisions at AP, we adopt a sequential ACK mechanism to allow receivers to return ACKs one by one. Sequential ACK is achieved by modifying the Network Allocation Vector (NAV). NAV is a duration field carried by IEEE 802.11 frames to provide virtual carrier sensing. NAV is used to reserve the medium for a fixed time period. The frame uses the NAV to reserve the medium for transmission sequence, i.e., the data frame, its acknowledgement, and the intervening SIFS. To ensure that the transmission sequence is not interrupted, a node sets the NAV in its frame (normally data frame or RTS frame) to block access to the medium while the frame is being transmitted. All nodes that hear the frame defer access to the medium until the NAV elapses. Each node maintains an NAV counter: when the NAV counts down to zero, the channel is idle; otherwise, the channel is busy. All nodes monitor the headers of all frames they hear and update the NAV counters accordingly.

In Carpool, we need to reserve medium for multiple ACKs. We denote the transmission time for data frame, ACK frame, SIFS as $t_{\rm payload}, t_{\rm ACK}, t_{\rm SIFS}$, respectively. If the number of receivers is N, the aggregated data frame sets its NAV as

$$NAV_{\rm data} = t_{\rm payload} + N \cdot (t_{\rm ACK} + t_{\rm SIFS}). \tag{1}$$

Then, all nodes hearing the frame defer their access to the channel for $NAV_{\rm data}$. Receivers return ACKs sequentially according to the order of the subframes: the receiver of the first subframe waits an SIFS interval to return an ACK, while the receiver of the second subframe waits an SIFS interval after the transmission of the first receiver's ACK, and so forth. The position of the receiver's subframe can be obtained after successfully decoding the frame.

To enable sequential ACK transmissions, each receiver automatically updates its NAV after the reception of the frame as follows:

$$NAV_{\rm i} = (i-1) \cdot (t_{\rm ACK} + t_{\rm SIFS}), \tag{2}$$

where i indicates the receiver of the ith subframe. The receiver of the ith subframe updates its NAV counter by $NAV_{\rm i}$ after the reception of the frame.

The NAV in sequential ACK is modified to indicate the end of the whole ACK sequence, that is, the jth ACK sets its NAV to $NAV_{\rm N-j+1}$, where N is the number of receivers and $NAV_{\rm i}$ is defined by Eq. (2). As such, the last ACK sets its NAV to $NAV_1 = 0$, which is consistent with the legacy ACK.

Note that it is possible that the AP only receives a subset of ACKs. In this case, the AP needs to identify the sources of the missing ACKs to determine which parts of the frame require retransmission. As differences in signal propagation and processing delays for multiple receivers are quite small, the AP can match ACKs to subframes by checking the receiving timestamps of the ACKs.

In dense environments, it is likely there exist hidden terminals, which cannot be coordinated by carrier sensing and may cause collisions in data frame and ACK transmissions. To mitigate hidden terminal issues, we adopt a mechanism based on the RTS/CTS signaling in IEEE 802.11. Analogous to multicast RTS/CTS [17], we employ a mechanism in which one multicast RTS is followed by a sequence of CTSs sent by receivers, as illustrated in Fig. 7. Different to multicast RTS/CTS, (i) we add an A-HDR, which is the same with the A-HDR in the data frame, to the RTS frame; (ii) the CTS frame format conforms to the legacy CTS, while NAV of each CTS is modified to indicate the end of the sequential ACK.

4.3 Backward Compatibility

So far we have presented the aggregation mechanism design of Carpool. A key factor that determines the acceptance of

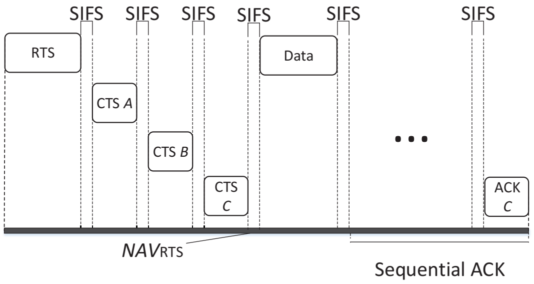


Fig. 7. Illustration of RTS/CTS mechanism in Carpool.

Carpool design is that whether it be able to compatible with legacy protocols. We answer this question affirmatively in the following three aspects.

AP Association. STAs indicate their supported protocols, including Carpool and versions of legacy protocols (e.g., IEEE 802.11b and 802.11n), to APs during association. When communicating with an STA, the AP runs the corresponding version of protocol supported by the client. In particular, AP may aggregate frames for multiple Carpool nodes using Carpool protocol, while sending frames to other legacy nodes using corresponding protocol versions.

Frame Decoding. Legacy nodes can perform legacy MAC frame aggregation and normal frame transmission when they coexist with Carpool nodes. Carpool frames are aggregated at PHY, and are compatible with legacy MAC frame aggregation and legacy non-aggregated frames. On the one hand, Carpool nodes can easily recognize Carpool frames and legacy frames by decoding A-HDR at PHY. On the other hand, legacy nodes does not support the PLCP of Carpool frames, and therefore cannot decode Carpool frames at PHY. Note that legacy MAC frame aggregation shares the same PHY frame format with a single legacy frame, and thus can be successfully decoded by legacy nodes without causing any confusion to Carpool nodes.

MAC Compatibility. Since Carpool still conforms to CSMA/CA at MAC layer, Carpool nodes can coexist with legacy nodes. Concretely, Carpool nodes precisely follow the MAC protocol specified by IEEE 802.11 to contend channel with other legacy nodes – Carpool nodes will defer if they sense a transmission of IEEE 802.11 nodes and vice versa.

5 Real-Time Channel Estimation

Based on the aggregation mechanism presented earlier, STAs can opt for the right fraction of the Carpool frame. To decode the right fraction, the STAs need to accurately estimate the channel condition during the transmission of that fraction. As observed in Section 3, the channel variance during the transmission of a long frame causes BER bias which compromises the reliability of long frame transmission. In this section, we propose a real-time channel estimation scheme to continuously calibrate channel information in the decoding process.

5.1 Calibrating Channel Estimation Using Data Pilots

The fundamental reason for BER bias is that the channel condition is measured at preamble while channel condition varies over a long frame transmission. To combat channel variance, channel estimation needs to be continuously calibrated based on the latest information. To achieve this goal, *the basic idea is to leverage correctly decoded data subcarriers to track channel variance.* The correctly decoded data subcarrier is referred to as *data pilot*. This intuition is simple, yet it is still challenging to realize it on existing OFDM frames. The existing frame structure only checks the correctness of the whole payload using cyclic redundancy check (CRC) at the end of a frame, while the correctness of a certain data symbol is unknown. To track channel variance at symbol granularity, symbol-level CRC is required. Carpool carries symbol-level CRC checksum in the phase offset side channel which adds no extra overhead and makes no modifications to data symbols.

Channel estimation calibration in Carpool utilizes the long training field (LTF) in the preamble as well as data pilots to track channel variance. Specifically, the estimation from data pilots is used to calibrate the initial estimation from the LTF. The LTF estimates the channel based on the received signals of the known training symbols [18]. Similarly, the channel can be estimated using data symbols. We denote $\hat{H}_o$, $\hat{H}_n$ as the channel estimation results obtained using the LTF and the nth data symbol, respectively. If the received data of the nth symbol, denoted as D_n, is decoded correctly, the receiver can infer the transmitted signal Y_n. Channel estimation using D_n follows the rule: $\hat{H}_n = \frac{D_n}{Y_n}$. If D_n is decoded correctly, we incorporate $\hat{H}_n$ with previous estimated value to updating the channel estimation. Specifically, we calibrate the LTF estimated value $\hat{H}_o$ by iteratively updating channel estimation according to the following rule:

$$\tilde{H}_n = \begin{cases} (\tilde{H}_{n-1} + \hat{H}_n)/2 & D_n \text{ correctly decoded} \\ \tilde{H}_{n-1} & \text{otherwise} \end{cases} \tag{3}$$

Note that $\tilde{H}_n$ is the calibrated channel estimation for the nth symbol.

5.2 Phase Offset Side Channel

Note that the above channel calibration scheme assumes symbol level CRC in OFDM-based frame. However, adding such redundancy will cost large overhead and require modifications on existing decoding algorithms. To address this challenge, we introduce a side channel in existing OFDM frame structure to carry CRC bits without affecting the data transmission and decoding performance.

In OFDM system, carrier frequency offset between transmitter and receiver results in phase shifting in received signals. Existing OFDM based Wi-Fi standards, e.g., IEEE 802.11a/g/n/ac, estimate and compensate for frequency offset using the short training field (STF) and LTF in preamble. Nevertheless, a residual frequency error might exist even after frequency correction has been applied. The phase shift caused by the residual frequency error grows proportionally with time and results in constellation rotation. To combat residual frequency error, OFDM based Wi-Fi standards insert several known carriers, referred to as *pilot subcarriers*, in each OFDM symbol. The pilot carriers in each OFDM symbol are considered as references to estimate phase offset in the

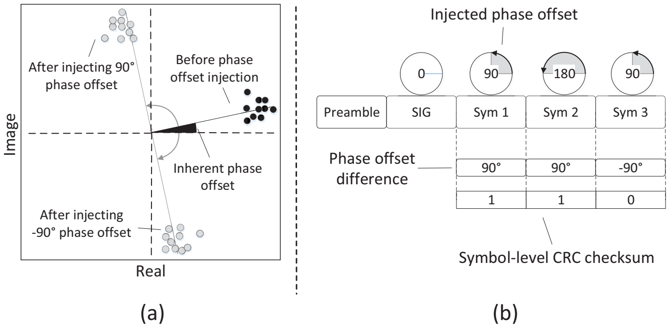


Fig. 8. Phase offset injection on OFDM symbol. (a) 90° phase shift on BPSK constellation diagram. (b) Phase offset difference between two consecutive symbols is used to convey bits.

data carriers of the OFDM symbol. Whereas, the four pilot subcarriers are not enough to track the channel variance during the transmission of a long frame, as indicated in Fig. 3.

Our goal is to leverage inherent phase tracking ability in IEEE 802.11 OFDM PHY to create a side channel without hindering data decoding. Specifically, the transmitter explicitly injects extra phase offset to each payload symbol after data modulation. No phase offset is injected into the preamble symbols so that the preamble can estimate frequency offset and channel condition. In each OFDM symbol, a phase offset is injected by rotating the complex constellation points of all data subcarriers and pilot subcarriers. As such, pilot and data subcarriers still maintain the same amount of total phase offset, which includes both inherent and injected phase offsets. Fig. 8(a) illustrates a BPSK constellation diagram with $\pm 90^\circ$ phase offset injections. In this figure, the total phase offset of the complex constellation points is comprised of inherent phase offset and injected phase offset. The receiver tracks the total phase offset using pilot subcarriers and compensates the total phase offset before data demodulation. As the pilot-based phase tracking accuracy is affected by channel estimation and noise but independent of the amount of phase offset, phase offset injection has no negative impacts on the pilot-based phase tracking. As we only inject phase offsets in payload symbols and the preamble remains unchanged, phase offset injection has no impacts on channel estimation and frequency offset estimation in the preamble. Therefore, data decoding is not affected by phase offset injection.

To encode bits into phase offset, we need to consider the inherent phase offset caused by residual frequency offset. As the inherent phase shift caused by residual frequency error accumulates across symbols, the inherent phase offsets can be very large in the symbols at the tail of a frame, making it challenging to decode injected phase offset directly from the pilot phase tracking results. To combat residual frequency offset, we observe that the difference between the inherent phase offsets of two consecutive symbols is quite small. Therefore, we use the difference between the phase offsets of two consecutive symbols to encode the side channel bits. As illustrated in Fig. 8(b), 90° and -90° are used to modulate “1” and “0”, respectively. To convey the bit sequence “110” over the three symbols in the figure, we need to keep the phase offset differences as 90°, 90°, and -90°, thereby injecting 90°, 180°, and 90° phase offsets into the three symbols, respectively. The modulation schemes used for phase offset side channel are summarized in Table 1. Normally, the phase offset modulation is more robust compared with corresponding phase-shift keying modulations (i.e., BPSK and QPSK). This is because each symbol uses four pilot subcarriers to track phase offset.

TABLE 1
Phase Offset Modulation.

One-bit phase offset	Data
90°	1
-90°	0

Two-bit phase offset	Data
45°	11
135°	01
-135°	00
-45°	10

The decoding procedures of data and phase offset are independent: the inherent phase tracking capability of OFDM receiver detects and compensates the total phase offset of each symbol before data decoding. To correctly decode data symbol, we only need to obtain the total phase offset rather than the injected phase offset, which means that even if phase offsets are not decoded correctly, it has no negative impacts on data decoding. This feature of phase offset side channel ensures that phase offset injections do not sacrifice robustness of data decoding, which is different from pilot side channels [19], [20] and other dirty coding side channels [21], [22].

The price of creating a phase offset side channel is that pilot subcarriers cannot estimate the inherent phase offset caused by frequency offset and channel. This information is used in some proposals for frequency offset calibration or channel estimation. However, in IEEE 802.11 OFDM PHY standards, pilot subcarriers are only used for phase tracking, whose accuracy is not affected by phase offset injection. Therefore, we claim that phase offset side channel does not affect data decoding performance in IEEE 802.11 standards.

Carpool leverages the phase offset side channel to add symbol-level CRC checksum checksum, which is used to indicate the correctness of a group of symbols. The group of symbol is considered as the basic unit for checking CRC and updating channel estimation. The key design question here is *what is the best granularity for symbol-level CRC*: i) phase offset side channel offers different modulation schemes to allow each symbol to carry different numbers of CRC bits, ii) and the number of symbols sharing a symbol-level CRC checksum controls the granularity of the channel estimation calibration. Ideally, we want as many the bits of CRC checksum as possible and as few symbols sharing a symbol-level CRC checksum as possible. Unfortunately, these goals are in conflict with each other. On the one hand, shorter length of CRC checksum compromises the reliability of the CRC itself. On the other hand, longer CRC checksum requires more symbols to carry, which offers less opportunity to update channel condition as one error symbol will drag down the whole symbol group, and thus there will be less

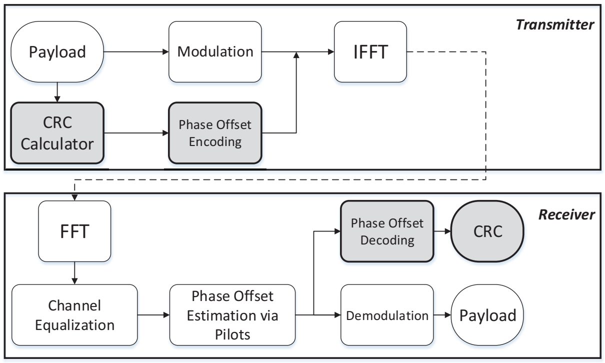


Fig. 9. Transceiver Architecture with Phase Offset Side Channel.

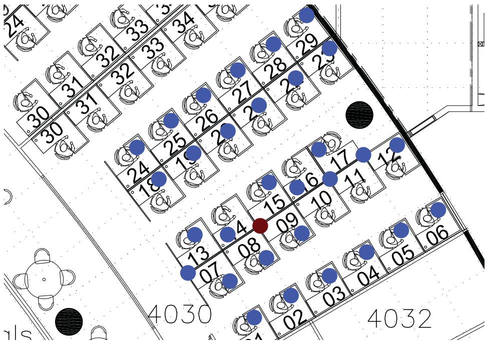


Fig. 10. Testbed layout. Carpool transmitter (red node) is placed in the center of the room. Carpool receiver (blue nodes) is placed in 30 different locations.

data pilots. To seek an optimal tradeoff between reliability and granularity, we conduct several measurements on the channel estimation performance using different modulation and granularity schemes in indoor environments. We vary the transmission power from 0.05 to 0.2 (in USRP's power magnitude unit), and place the receiver in 30 different locations in the same office with the transmitter. We test the performance using six different schemes: two modulation schemes for phase offset side channel as listed in Table 1 combined with 1-3 symbols as a group for CRC check. The results show that the scheme with one symbol as a group and two-bit phase offset side channel achieves best performance in most the cases. The two-bit phase offset side channel is demonstrated to be very robust, and CRC-2 for each symbol offers a good tradeoff between reliability and granularity.

6 Implementation

Carpool can be realized in the existing OFDM PHY with no change in hardware. We have implemented the basic mechanisms of Carpool atop the OFDM implementation of the GNURadio/USRP platform. We implement the entire PHY design specified in Section 3 directly in the USRP Hardware Drive (UHD). Nodes in our experiments are equipped with R-FX2450 daughterboards as RF frontend, which are configured to operate in the 2.4-2.5GHz range.

Carpool adopts the legacy PHY layer convergence procedure (PLCP) format of IEEE 802.11n, in which PLCP preamble consists of two OFDM symbols for the short training field (STF) and two OFDM symbols for the LTF, and PLCP header consists of one OFDM symbol for SIG. For Carpool, two symbols between PLCP preamble and PLCP header are used for A-HDR. Each OFDM symbol consists of four pilot subcarriers and 48 data subcarriers. The frame synchronization and channel equalization algorithms are implemented according to IEEE 802.11n.

Figure 9 depicts the implementation of phase offset side channel. The phase offset encoding process is implemented by adding two extra blocks on standard PHY: symbol-level CRC calculator and phase offset encoder. The symbol-level CRC calculator computes CRC checksum for each symbol before the raw bits are mapped to the constellation. Phase offset encoder modulates CRC bits into phase offsets using the modulation described in Section 5.2. Before feeding a payload symbol into IFFT, the symbol is rotated based on the phase offset. Phase shifting of an angle θ is realized in the frequency domain by multiplying data signals with $e^{j\theta}$. The decoding is implemented by reversing the encoding procedure. We have added the phase offset decoder block to the receiver chain. Phase rotation is tracked in the frequency domain using existing blocks in the receiver chain. The extracted phase rotations are fed into the phase offset decoder to obtain the phase offset bits. The phase offset decoded at the receiver is considered to fall into $(-180°, 180°]$. As we adopt the phase offset difference between two consecutive symbols for encoding, it does not cause ambiguity even if the injected phase offset rotation plus the original phase offset exceeds the range of $(-180°, 180°]$. The whole data decoding procedure in the receiver chain – including channel equalization, phase offset compensation – is unmodified.

7 System Evaluation

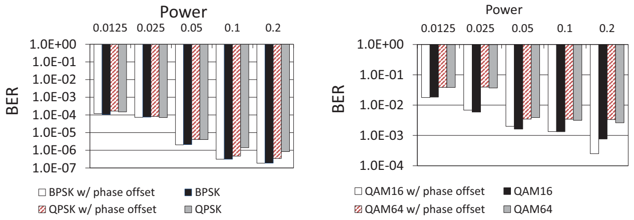
We evaluate Carpool in this section. We first evaluate the PHY implementation in Section 7.1. Then, in Section 7.2, we conduct MAC evaluation using trace-driven simulations in a relatively large networking setting.

7.1 PHY Evaluation

We evaluate the feasibility of Carpool using our PHY implementation on USRP. We conduct testbed experiments using three USRP nodes: one as transmitter and the other two as receivers. The USRP nodes are placed at different locations in a $10m \times 10m$ office as depicted in Figure 10.

7.1.1 Evaluation of Phase Offset Side Channel

Impact on data decoding. We evaluate the impact of phase offset side channel on data decoding by comparing the single link performance of two PHYs: the standard PHY without phase offset side channel and the PHY with phase offset side channel. We conduct experiments in a controlled environment: we use identical and static indoor layouts for the two schemes. Two-bit phase offset is adopted to convey CRC checksum of each symbol. We use standard channel estimation algorithm at the receiver for both PHYs. The power magnitude is



(a) BPSK and QPSK (b) QAM16 and QAM64

Fig. 11. BER of PHY with phase offset side channel vs the standard PHY.

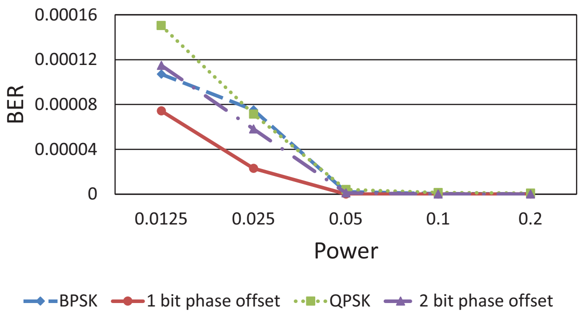


Fig. 12. BER of phase offset side channel vs data channel.

the transmission gain set through UHD with respect to the maximum power of the XCVR2450 daughterboard, which is 20dBm.

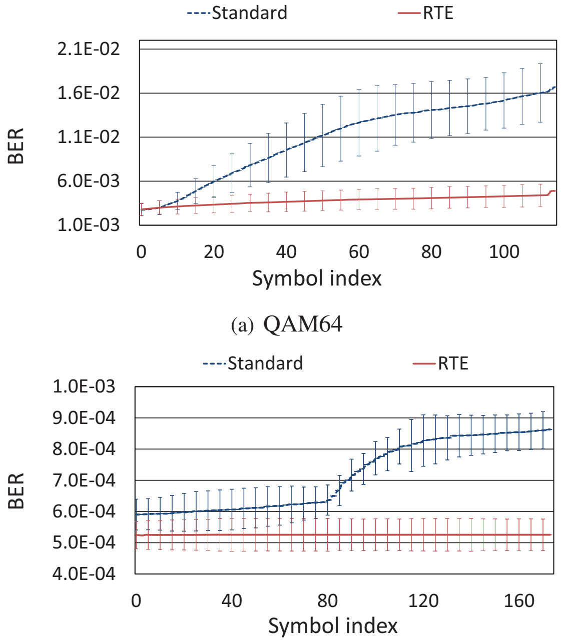
Fig. 11 compares the BER performance of the two PHYs with different modulations. Across all modulation schemes demonstrated, the BER differences of two PHYs range from 1.02% to 5.49%, which shows that introducing phase offset side channel has little impact on data decoding performance.

Reliability. We evaluate the reliability of phase offset side channel by comparing the BER performance of phase offset side channel with OFDM data channel. We vary the transmission power to test the BER performance under different SNR environments. In each power setting, we send 1KB frames with phase offset side channel, and compute the BERs of phase offset bits and payload.

Figure 12 shows the BER performance of phase offset side channel compared with data subcarrier. Figure 12 shows that 1-bit phase offset has better BER performance compared to BPSK, and the BER of 2-bit phase offset is much lower than QPSK in most cases. This is because phase offset side channel uses four pilot subcarriers to demodulate one phase offset, while bits carried on each data subcarrier are demodulated independently.

7.1.2 Channel Estimation Performance

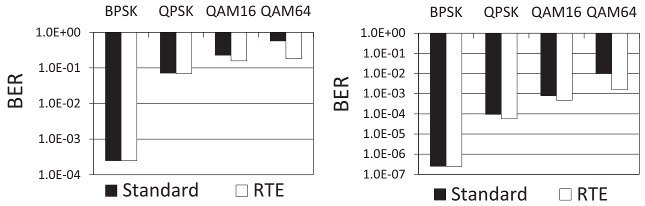
In Fig. 13 and Fig. 14, we evaluate the real-time channel estimation (RTE) scheme proposed in Section 5. The transmitter sends 4KB frames in 2M channel, in which the transmission time of a 4KB frame corresponds to a 40KB frame in 20M channel. To compare the two channel estimation schemes, the receivers log the frames, and then decode the frames offline using the two schemes, respectively.



(a) QAM64

(b) QAM16

Fig. 13. BER bias of RTE vs standard.



(a) Power magnitude $= 0.05$ (b) Power magnitude $= 0.2$

Fig. 14. BER performance of RTE vs standard.

In Fig. 13, we set transmission power to 0.2 and vary the locations of the receivers. The error bars stand for standard deviation. The results show that RTE largely eliminates the BER bias and maintains relatively low BER even at the tail of a frame. For example, the BER at the 100th symbol of QAM64 modulated frame is lower than 5×10^{-3} using RTE, while the corresponding BER using the standard channel estimation is higher than 1.5×10^{-2}. Compared to the standard channel estimation, RTE reduces the BER of of QAM64 amd QAM16 modulated frames by 65% and 27%, respectively.

Fig. 14 evaluates the BER performance of real-time channel estimation for different modulation schemes. We observe that for higher-order modulation schemes, i.e., QAM16 and QAM64, real-time channel estimation achieves several times lower BERs compared to traditional channel estimation, while for lower-order modulation schemes – BPSK and QPSK – the BER gains achieved by real-time channel estimation are marginal. The reason is that higher-order modulation schemes are more error-prone to channel variance.

TABLE 2
PHY/MAC parameters.

Slot time	$9\mu s$
SIFS	$10\mu s$
DIFS	$28\mu s$
Minimal contention window	15 time slots
Maximal contention window	1023 time slots
PLCP header	$28\mu s$
Propagation delay	$1\mu s$

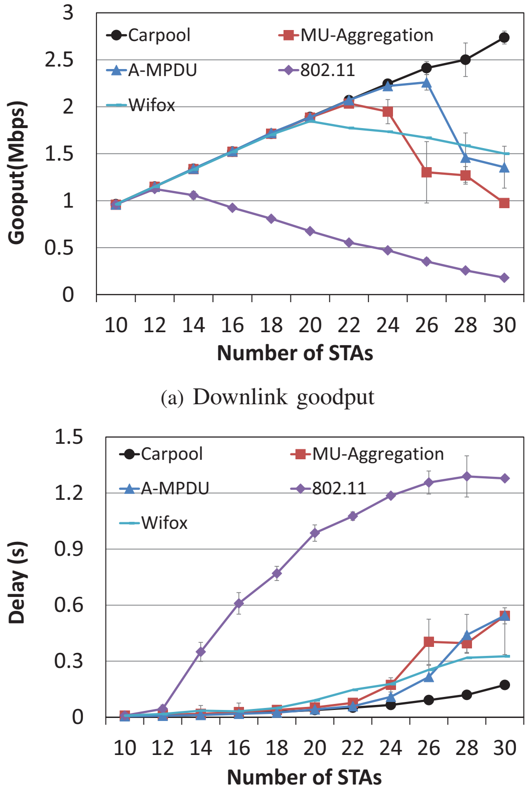
7.2 MAC Evaluation

In this section, we evaluate the MAC performance of Carpool in large audience environments. Due to the processing delay, USRP2 cannot support real-time MAC layer protocols. Thus, we turn to trace-driven simulations in this section to evaluate the performance of Carpool. To conduct high fidelity emulation of real world settings, we feed empirical channel and traffic traces into the simulator to run MAC protocols.

7.2.1 Simulation Methodology

We have implemented a custom event-driven simulator using MATLAB. In the simulations, all nodes, including two APs and a variable number of STAs are within carrier sense range. The number of STAs ranges from 10 to 30, which is a typical range as we measured in the library WLAN. We use the values of SIFS, DIFS, and exponential backoff contention as in IEEE 802.11n. Table 2 summarizes the parameters used in our emulation. We emulate the frame decoding performance based on the traces collected from USRP nodes. The trace-driven emulation is performed as follows. We select the least busy channel in our office to minimize uncontrollable interference from neighboring APs. We place the transmitter at the center of the office and vary the locations of receivers to act as different STAs. The locations are depicted in Fig. 10. The transmitter continuously sending frames to receivers. We vary frame lengths to match those in our simulation. The traces at each location are fed into one STA in our simulation. We log the traces at USRP and decode frames offline using the standard channel estimation and real-time channel estimation, respectively.

To demonstrate the merits of Carpool, we also implement the MPDU aggregation (A-MPDU) as defined in IEEE 802.11n [23], multi-user aggregation (MU-Aggregation) without RTE [8], [9], and WiFox [3]. A-MPDU aggregates frames that are buffered at AP for one STA. MU-Aggregation relaxes the constraint of one STA by extending aggregation to multiple STA. Carpool differs from these approaches in the following aspects: i) Carpool employs the RTE channel estimation for long aggregated frames while both A-MPDU and MU-Aggregation adopt standard channel estimation, and ii) both Carpool and MU-Aggregation allow aggregation for multiple receivers while A-MPDU restricts aggregation to one STA. Therefore, comparison with A-MPDU shows the advantages of aggregation for multiple STAs in large audience environments, and comparison with MU-Aggregation demonstrates the benefits of employing RTE in long aggregated frames. WiFox alleviates traffic asymmetry issue by giving higher priority to downlink transmission in channel contention. Comparison with WiFox shows the merits of our PHY design.



(a) Downlink goodput

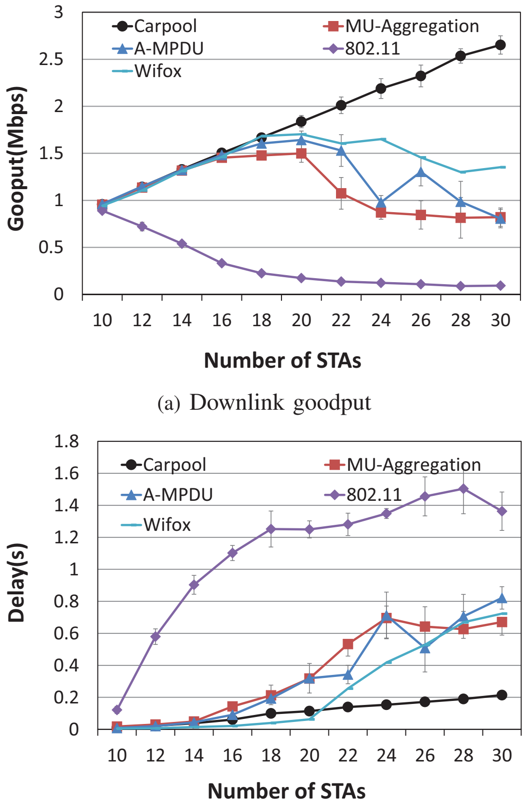
(b) Downlink latency

Fig. 15. Goodput and latency performance for VoIP traffic.

7.2.2 MAC Performance

VoIP traffic. We first evaluate the performance of Carpool with delay-sensitive VoIP traffic, which is an ON/OFF UDP stream with a peak rate of 96Kbit/s and frame size of 120B according to IEEE 802.11n requirements [24]. Normally, VoIP is considered as a challenging application for aggregation schemes due to its ON/OFF nature and small frame size. We adopt Brady's model [25] to generate VoIP traffic. The PHY data rate is 65Mbit/s.

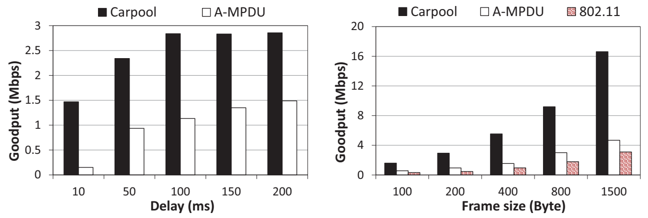
Fig. 15 compares Carpool with IEEE 802.11n with a various number of STAs. We make several observations. First, Carpool outperforms IEEE 802.11 with and without aggregation in terms of throughput and latency when the number of STAs associated with one AP is large. As shown in Fig. 15(a), when the number of STAs increases from 22 to 30, the overall downlink goodput of Carpool keeps increasing linearly, while the goodput of A-MSDU tapers off quickly from 2Mbit/s to about 1Mbit/s, and the goodput of IEEE 802.11 drops quickly from 0.55Mbit/s to 0.18Mbit/s. MU-Aggregation achieves slightly lower goodput compared with A-MPDU, and is significantly outperformed by Carpool. This is because Carpool reduces contention by aggregating more frames, and the performance gain of sending long frames heavily relies on reliable transmission guaranteed by RTE. WiFox achieves



(a) Downlink goodput

(b) Downlink latency

Fig. 16. Goodput and latency performance with UDP/TCP traffic in background.



(a) Goodput performance for different latency requirements (b) Goodput performance for different frame sizes

Fig. 17. Goodput performance for different latency requirements and frame sizes.

higher goodput than IEEE 802.11 when numbers of STAs are large. This is because WiFox alleviates traffic asymmetry issue by giving higher priority to downlink traffic. Carpool still outperforms WiFox since Carpool reduces contention by providing reliable long transmissions.

Fig. 15(b) shows that when the number of STAs increases, the latency of Carpool remains very low compared with other schemes. Additional STAs leads to increased contention and likelihood of collisions, which hinders the downlink goodput. Carpool reduces contention by sending traffic for different STAs in a single transmission. Due to the ON/OFF pattern and small frame size of VoIP traffic, A-MSDU can only aggregate a limited number of frames for a single STA, which is insufficient when the number of contenders is large.

UDP/TCP traffic in SIGCOMM trace. We evaluate the performance of Carpool in presence of uplink background traffic in busy networks. The motivation for this evaluation is that as reported in [3], [5], the traffic analysis on SIGCOMM'08 trace shows that a major cause of poor performance in public WLANs is uplink background traffic. We inject UDP/TCP traffic according to SIGCOMM'08 trace [5], where the average inter-packet arrival times for TCP and UDP are 47ms and 88ms, respectively. The frame size distribution of the SIGCOMM'08 trace is depicted in Fig. 1(b).

Fig. 16 depicts the downlink performance of Carpool and IEEE 802.11 schemes. Compared to the pure VoIP traffic scenario, we can see that uplink traffic has dragged down the throughput, especially for benchmark approaches. The reason is that TCP/UDP background traffic makes contention more intense. The performance of Carpool is less affected by background traffic Carpool has plenty of opportunities in busy networks to reduce contention by reliably aggregating traffic for multiple STAs. From Fig. 16(a), we observe that when the number of STAs increases from 20 to 30, Carpool achieves from 1.12 to 3.2 times the goodput of A-MPDU. As shown in Fig. 16(b), the latency of Carpool remains lower than 0.2s, while A-MPDU and IEEE 802.11 suffers the worse delays of 0.8s and 1.5s, respectively. It is worth noting that when the number of STAs is less than 10, delays of all approaches are almost zero, which reveals that the settings with less than 10 STAs are not congested. This is because traffic is not saturated under such settings, and all packets are delivered without getting congested at APs. In these cases, Carpool has few opportunities to aggregate frames, and thus transmits single frames in the same manner with IEEE 802.11.

Performance with different latency requirements and frame sizes. We evaluate the benefits of Carpool under different latency and frame size conditions. The aggregation process is ended when the size of the buffered frames reaches the maximum frame size or the delay of the oldest frame reaches the maximum latency limit. Unless otherwise stated, we use the same settings as in Fig. 16. We focus on MAC layer performance and assume frame retransmission is only caused by collision. The same uplink background traffic is injected as in Fig. 16, while we set the downlink traffic with specific latency requirements and frame sizes to replace VoIP traffic. The number of STAs associated with each AP is fixed to 30.

In Fig. 17(a), we test the goodput of VoIP traffic with different latency requirements. Carpool achieves 1.9-9.8× goodput compared to A-MPDU. The goodput gain of Carpool is smaller as the latency bound is looser. This is because the goodput of Carpool approaches the overall traffic arrival rate, while A-MPDU still has congested traffic to aggregate.

Fig. 17(b) keeps the latency requirements at 10ms and vary the frame sizes from 100B to 1500B. Fig. 17(b) shows that Carpool achieves considerable goodput gain by a factor of between 2.8 and 3.6 over A-MPDU, and achieves 5 to 6.4× goodput gain compared with IEEE 802.11. With larger frame sizes, Carpool still achieves similar goodput gain over IEEE 802.11 schemes, as the benefits of Carpool comes from aggregation for multiple STAs which remains the same unless

Carpool frame reaches 64K.

8 Discussion

Fairness. Carpool allows APs to aggregate multiple frames into a single transmission, which renders potentially higher priority to downlink traffic. Similarly, as downlink traffic dominates the overall traffic in WLANs, state-of-the-art scheduling mechanisms [3], [6] address this issue by offering higher priority to downlink traffic. Therefore, Carpool ensures heavy-loaded APs higher priority, which follows the notion of fairness that existing priority control schemes strive to accomplish.

For the fairness among multiple receivers of one AP, the design of Carpool is orthogonal to scheduling and fairness control schemes, which can be built on top of Carpool. In this paper, the priority control of delay insensitive downlink traffic implemented in each AP is first in first out (FIFO), and delay sensitive traffic is assigned with higher priority. Other priority and fairness control schemes such as time fairness control can also be implemented on Carpool by maintaining a time occupancy table for all STAs. The scheduling module in AP periodically checks the time occupancy table and assigns higher priority to STAs with smaller time occupancy.

Processing latency. The PHY design of Carpool causes extra computations in baseband, which is mainly contributed by the A-HDR generation/check and the phase offset side channel encoding/decoding. Carpool generates and checks the A-HDR based on the insertion and check operations of Bloom Filter, which possesses the appealing feature that the time needed for insertion or check is in the set is a fixed constant $O(h)$ (h is the number of hash functions) that is completely independent of the number of items in the Bloom Filter. Normally, the A-HDR generation or check takes merely no more than a few micro seconds with a typical digital signal processor [26]. The phase offset side channel encoding/decoding is processed in parallel with data encoding/decoding in our implementation, as shown in Fig. 9. As the number of bits in the phase offset side channel is one or two orders of magnitude smaller than that of payload, the processing latency caused by phase offset side channel is negligible. These delays could be further reduced in future chip designs.

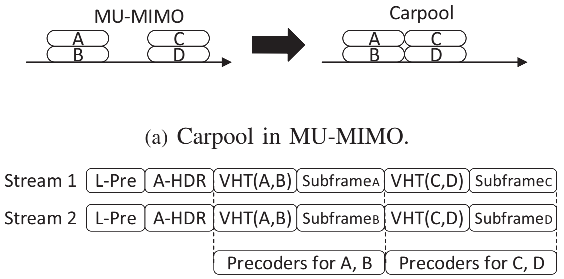
Energy consumption. As Carpool aggregates frames from multiple receivers, the frame decoding takes longer time and consumes more energy. As a receiver does not need to decode all subframes, this problem can be mitigated by just sampling without decoding, or downclocking the sampling rate like [27]. The sampling rate switching delay is merely $9.5\mu s$, and thus Carpool receivers can immediately downclock their sampling rates after receiving their intended subframes. In addition, if a Carpool receiver checks the A-HDR and finds that all the subframes are not intended for it, it can directly enter idle mode without decoding any subframe. As there are certain false positives of decoding A-HDR, Carpool consumes more energy when a node decodes irrelevant frames due to false positives. Given the lack of real Wi-Fi chipsets implemented with Carpool to measure power consumption, we use a device power model to estimate the power consumption of Carpool based on its transmission behaviors. We select a device power model from [27], which is based on measurements of LinkSys WPC55AG NIC. In the power model, the mean power consumptions of Wi-Fi under transmission (TX), receiving (RX), and idle (IL) modes are 1.71W, 1.66W, 1.22W, respectively. Recall that if the aggregated receivers are limited to 8, the false positive ratio is upper-bounded by 5.59%. This means that Carpool consumes at most 5.59% more RX power than standard Wi-Fi node. According to [27], in a busy network, for more than 92% of clients, 90% of energy is spent in IL, while TX and RX shares almost the same amount of energy cost. As such, for more than 92% of clients, a Carpool node spent at most $5.59\% \times 5\% = 0.28\%$ more energy than a standard Wi-Fi node.

It is worth noting that Carpool achieves up to $3.2\times$ goodput gain, which shortens the communication time for given amount of traffic. Therefore, Carpool nodes have more time left to enter power save mode (PSM) as specified by IEEE 802.11 to save more energy.

Extension on MIMO. The implementation of Carpool is based on IEEE 802.11a. We now briefly discuss how to implement Carpool in the MIMO cases in IEEE 802.11n/ac. We first present the extension of Carpool in Multi-user MIMO (MU-MIMO) in IEEE 802.11ac, and then show that it also can be applied to single-user MIMO in 802.11n. On the one hand, IEEE 802.11ac supports up to 8 spatial streams and the number of spatial streams in MU-MIMO is limited by the number of antennas that have been equipped in the AP, which is still not enough to support concurrent downlink traffic for scores of STAs in public WLANs [5]. On the other hand, IEEE 802.11ac allows higher PHY data rate which further reduces MAC efficiency. Thus, aggregating more downlink traffic in a single transmission in MU-MIMO is still desirable.

Carpool can be extended to MU-MIMO with simple modifications on precoders. Carpool aggregates multiple beamformed streams by sharing a single legacy preamble while keeping their own VHT preambles and precoders. Figure 18(a) illustrates the aggregation scheme for MU-MIMO. Suppose a two-antenna AP has four data streams for four different STAs. IEEE 802.11ac MU-MIMO requires at least two transmissions where each transmission contains two data streams. Carpool can aggregate four data streams into one transmission by using the frame structure depicted in Figure 18(b). Similar to previous design, the A-HDR is added right after the legacy preamble. The Bloom filter contains information for four STAs: the indices of A, B are 1, and the indices of C, D are 2. VHT preamble and payloads for A, B are pre-coded by the precoder that is computed based on the channel estimation for A, B, while VHT preamble and payloads for C, D are pre-coded by the precoder computed based on the channel estimation for C, D. In this way, all STAs share a single preamble and Bloom filter and can decode their own payloads. The above MU-MIMO Carpool can also be applied to single-user MIMO. The only difference is that in single-user MIMO, different streams corresponding to one precoder are restricted to one receiver.

Implementation and deployment in practice. Future Wi-Fi standards such as High Efficiency WLAN (HEW) have already taken dense networks as a new scenario. New stan-



(a) Carpool in MU-MIMO.

(b) Carpool frame structure.

Fig. 18. Illustration of Carpool in MU-MIMO.

dards will be proposed for this scenario, and may include new PHY/MAC specifications. Hereby we discuss how Carpool can be implemented in practice to benefit future Wi-Fi networks. PHY changes we made in Carpool can be easily implemented on today's hardware with only software and frame format update. Note that we can re-use existing functions in PHY to realize Carpool, and therefore today's hardware is enough to implement Carpool. Thus, vendors can incorporate Carpool into Wi-Fi nodes by developing a new firmware that realize the function of Carpool using existing PHY function blocks and computational logics.

9 Related Work

Dense Wi-Fi Networks. The scenario of dense Wi-Fi networks have been intensively discussed by both standardization bodies and researchers in recent years [1]. The initial motivation of dense Wi-Fi networks is to meet the increasing Internet access demands of mobile devices in dense public areas, while the massive numbers of connectivities and huge data traffic demands overturn many design principles that worked well for traditional Wi-Fi networks. Baid et al. [2] investigate the impact of increasing density of APs with centralized channel assignment on network throughput. Shin et al. [1] propose a dynamic sensing and power control mechanism to mitigate interference in dense Wi-Fi networks. Wang et al. [28] improve the MAC efficiency in dense WLANs in frequency domain by facilitating per-frame spectrum adaptation. Rahul et al. [29] propose a practical system to implement network MIMO on unmodified IEEE 802.11n cards to scale throughput with increasing number of users. Different from these studies, we focus on the MAC inefficiency and traffic asymmetry issues in Wi-Fi networks with dense users.

MAC frame aggregation. MAC frame aggregation is proposed in IEEE 802.11n [23] to aggregate MAC protocol data unit (A-MPDU) or MAC protocol service unit (A-MSDU) at MAY layer. Along with the standard MAC aggregation schemes, many works [9], [12] have proposed MAC designs for one-to-one aggregation, but will cause severe delay in real-time applications or short data flows. There are also several proposals for multi-user aggregation [8], [9], [13]. These approaches are limited to frame format design, while practical issues for large audience environments including BER bias and aggregation overhead are not yet been considered.

WLANs performance enhancement. A major solution for the large audience environment scenario is enterprise WLANs. The enterprise WLANs leverage centralized coordination to reduce unnecessary collisions and contention. CENTAUR [7] uses a selective amount of data path centralization to avoid potential interference. Centralized channel assignments for different links are explored in [6], [30] to combat hidden and exposed terminals. WiFox [3] analyzes the traffic patterns of large audience environments based on SIGCOMM'08 trace [5] and proposes a scheduling scheme to adaptively prioritize AP's channel access over competing STAs. These solutions have focused on higher-level resource allocation and scheduling, while Carpool tries to cope with the inefficiency issue by reducing contention overhead. Specifically, these solutions can be applied on top of Carpool to improve the performance of large scale WLANs.

Narrow channels. To improve MAC efficiency, a current trend in the research community is to split a wide channel into multiple narrow channels. FICA [16] proposes a synchronous system to use a sub-channel instead of a whole wide band channel for each transmission pair. FICA requires all transmitters in a carrier sense range to transmit within a few microseconds of each other. WiFi-NC [31] uses sharp elliptic filters to enable independent narrow channel access. The cost of using sharp elliptic filters is more guard bands or reduced SNR, as shown in [32]. Carpool is orthogonal to narrow channel designs and can be applied to narrow channel systems to further improve MAC efficiency and reduce collisions.

Pilot side channels. Several approaches [19], [20] utilize pilot subcarriers to convey side information, which is related to our phase offset side channel. The fundamental difference between phase offset side channel and existing pilot subcarrier side channels is that phase offset does not affect on the function of pilot in OFDM systems, and has no impact on data decoding.

10 Conclusion

In this paper, we characterize the causes of Wi-Fi inefficiency in large audience environments and propose Carpool to cope with the issue by enabling frame aggregation for multiple STAs in the downlink transmission. Carpool is built on the heels of a lightweight frame structure and a dynamic channel estimation scheme, and can be easily integrated into the existing Wi-Fi standards as an optional mechanism that is enabled in large audience environments to ease heavy contention.

We have prototyped Carpool using the GNURadio/USRP platform. The experimental results show that Carpool can achieve up to $3.2\times$ goodput and reduce 75% delay compared to the IEEE 802.11n aggregation.

Acknowledgement

The research was supported in part by grants from 973 project 2013CB329006, China NSFC under Grant 61173156, RGC under the contracts CERG 622613, 16212714, HKUST6/CRF/12R, and M-HKUST609/13, as well as the grant from Huawei-HKUST joint lab.

References

[1] K. Shin, I. Park, J. Hong, D. Har, and D.-H. Cho, “Per-node throughput enhancement in wi-fi densenets,” *IEEE Commun. Mag.*, vol. 53, no. 1, pp. 118–125, 2015.

[2] A. Baid and D. Raychaudhuri, “Understanding channel selection dynamics in dense wi-fi networks,” *IEEE Commun. Mag.*, vol. 53, no. 1, pp. 110–117, 2015.

[3] A. Gupta, J. Min, and I. Rhee, “WiFox: scaling WiFi performance for large audience environments,” in *Proc. ACM CoNEXT*, 2012.

[4] M. Rodrig, C. Reis, R. Mahajan, D. Wetherall, and J. Zahorjan, “Measurement-based characterization of 802.11 in a hotspot setting,” in *Proc. ACM SIGCOMM workshop on Experimental approaches to wireless network design and analysis*, 2005.

[5] A. Schulman, D. Levin, and N. Spring, “On the fidelity of 802.11 packet traces,” in *PAM*, 2008.

[6] S. Rayanchu, V. Shrivastava, S. Banerjee, and R. Chandra, “FLUID: improving throughputs in enterprise wireless lans through flexible channelization,” in *Proc. ACM MobiCom*, 2011.

[7] V. Shrivastava, N. Ahmed, S. Rayanchu, S. Banerjee, S. Keshav, K. Papagiannaki, and A. Mishra, “CENTAUR: realizing the full potential of centralized wlans through a hybrid data path,” in *Proc. ACM MobiCom*, 2009.

[8] B. Otal, J. Habetha, F. Dalmases, P. Li, M. Ghosh, and P. Garg, “Muliple receiver aggregation (MRA) with different data rates for IEEE 802.11n.” US Patent App. 11/569,039, 2005.

[9] Y. Xiao, “IEEE 802.11n: enhancements for higher throughput in wireless LANs,” *IEEE Wireless Commun. Mag.*, vol. 12, no. 6, 2005.

[10] M. Vutukuru, H. Balakrishnan, and K. Jamieson, “Cross-layer wireless bit rate adaptation,” in *Proc. ACM SIGCOMM*, 2009.

[11] A. Broder and M. Mitzenmacher, “Network applications of bloom filters: A survey,” *Internet mathematics*, vol. 1, no. 4, 2004.

[12] T. Li, Q. Ni, D. Malone, D. Leith, Y. Xiao, and T. Turletti, “Aggregation with fragment retransmission for very high-speed WLANs,” *IEEE/ACM Trans. Netw.*, vol. 17, no. 2, 2009.

[13] K. Lee, S. Yun, and H. Kim, “Boosting video capacity of IEEE 802.11n through multiple receiver frame aggregation,” in *Proc. IEEE VTC*, 2008, pp. 2587–2591.

[14] F. Chang, W.-C. Feng, and K. Li, “Approximate caches for packet classification,” in *Proc. IEEE INFOCOM*, 2004.

[15] F. Hao, M. Kodialam, T. Lakshman, and H. Song, “Fast multiset membership testing using combinatorial bloom filters,” in *Proc. IEEE INFOCOM*, 2009.

[16] K. Tan, J. Fang, Y. Zhang, S. Chen, L. Shi, J. Zhang, and Y. Zhang, “Fine-grained channel access in wireless LAN,” in *Proc. ACM SIGCOMM*, 2010.

[17] C. Campolo, C. Casetti, C.-F. Chiasserini, and A. Molinaro, “A multirate mac protocol for reliable multicast in multihop wireless networks,” *Elsevier Comput. Netw.*, vol. 56, no. 5, pp. 1554–1567, 2012.

[18] J. Terry and J. Heiskala, *OFDM Wireless LANs: A theoretical and practical guide*. Sams Publishing, 2002.

[19] J. Zhang, H. Shen, K. Tan, R. Chandra, Y. Zhang, and Q. Zhang, “Frame retransmissions considered harmful: improving spectrum efficiency using micro-ACKs,” in *Proc. ACM MobiCom*, 2012.

[20] W. J. Lee, L. Wang, D. Yoon, S. K. Park, and W. Kim, “Transmission of side information using pilot tones for OFDM systems over fading channels,” in *Proc. IEEE ICCS*, 2008.

[21] K. Wu, H. Tan, Y. Liu, J. Zhang, Q. Zhang, and L. Ni, “Side channel: bits over interference,” in *Proc. ACM MobiCom*, 2010.

[22] A. Cidon, K. Nagaraj, S. Katti, and P. Viswanath, “Flashback: decoupled lightweight wireless control,” *Proc. ACM SIGCOMM*, 2012.

[23] “Part 11: Wireless LAN medium access control (MAC)and physical layer (PHY) specifications amendment 5: Enhancements for higher throughput,” *IEEE Std 802.11n-2009*, 2009.

[24] A. Stephens and et al., “IEEE P802. 11 wireless LANs: usage models,” *IEEE 802.11-03/802r23*, 2004.

[25] W. Wang, S. C. Liew, and V. O. Li, “Solutions to performance problems in VoIP over a 802.11 wireless LAN,” *IEEE Trans. Veh. Technol.*, vol. 54, no. 1, 2005.

[26] B. Debnath, S. Sengupta, J. Li, D. J. Lilja, and D. H.-C. Du, “Bloomflash: Bloom filter on flash-based storage,” in *Proc. IEEE ICDCS*, 2011.

[27] X. Zhang and K. G. Shin, “E-mili: Energy-minimizing idle listening in wireless networks,” in *Proc. ACM MobiCom*, 2011.

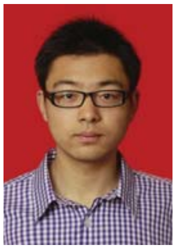
[28] W. Wang, Y. Chen, Z. Wang, J. Zhang, K. Wu, and Q. Zhang, “Changing channel without strings: Coordination-free wideband spectrum adaptation,” in *Proc. IEEE INFOCOM*, 2015, pp. 1–9.

[29] H. S. Rahul, S. Kumar, and D. Katabi, “Jmb: scaling wireless capacity with user demands,” in *Proc. ACM SIGCOMM*, 2012, pp. 235–246.

[30] R. Murty, J. Padhye, R. Chandra, A. Wolman, and B. Zill, “Designing high performance enterprise Wi-Fi networks,” in *Proc. USENIX NSDI*, 2008.

[31] K. Chintalapudi, B. Radunovic, V. Balan, M. Buettener, S. Yerramalli, V. Navda, and R. Ramjee, “WiFi-NC: WiFi over narrow channels,” in *Proc. USENIX NSDI*, 2012.

[32] S. Yun, D. Kim, and L. Qiu, “Fine-grained spectrum adaptation in WiFi networks,” in *Proc. ACM MobiCom*, 2013.



Wei Wang (S’10) is currently a Research Assistant Professor in Fok Ying Tung Graduate School, Hong Kong University of Science and Technology (HKUST). He received his Ph.D. degree in Department of Computer Science and Engineering from HKUST. Before he joined HKUST, he received his bachelor degree in Electronics and Information Engineering from Huazhong University of Science and Technology, Hubei, China, in June 2010. His research interests include privacy and fault management in wireless networks.

Yingjie Chen received his M.Phil. degree of computer science department from Hong Kong University of Science and Technology in 2012. He is currently a reseach assistant in Hong Kong University of Science and Technology. His research interests include PHY and MAC layer design in Wi-Fi network, and mobile computing.

Qian Zhang (M’00-SM’04-F’12) joined Hong Kong University of Science and Technology in Sept. 2005 where she is a full Professor in the Department of Computer Science and Engineering. Before that, she was in Microsoft Research Asia, Beijing, from July 1999, where she was the research manager of the Wireless and Networking Group. She is a Fellow of IEEE for “contribution to the mobility and spectrum management of wireless networks and mobile communications”. Dr. Zhang received the B.S., M.S., and Ph.D. degrees from Wuhan University, China, in 1994, 1996, and 1999, respectively, all in computer science.

Kaishun Wu (S’08-M’11) received the PhD degree in computer science and engineering from the Hong Kong University of Science and Technology (HKUST) in 2011. He is currently a research assistant professor at the Fok Ying Tung Graduate School at HKUST. His research interests include wireless communication, mobile computing, wireless sensor networks, and data center networks. He is a member of the IEEE and the IEEE Computer Society.

Jin Zhang (S’06-M’09) is currently an assistant professor in Electrical and Electronic Department, South University of Science and Technology of China. She graduated from Department of Electronic Engineering at Tsinghua University in 2004 with a bachelor’s degree and in 2006 with a master’s degree. She received the Ph.D. degree from Department of Computer Science and Engineering, Hong Kong University of Science and Technology. Her research interests are mainly in next-generation wireless networks, network economics, mobile computing in healthcare, cooperative communication and networks.