

Retransmission \neq Repeat: Simple Retransmission Permutation Can Resolve Overlapping Channel Collisions

Li Erran Li* Junliang Liu† Kun Tan†
Harish Viswanathan* Yang Richard Yang×

ABSTRACT

Collisions in overlapping channels are becoming an increasingly important problem in the deployment of high-speed wireless networks. In this paper, we present Remap, a simple, novel paradigm for handling collisions in overlapping OFDM channels. Remap introduces the novel concept of retransmission permutation that permutes the bit-to-subcarrier assignment after each retransmission, departing from the traditional, simply-repeat paradigm. Remap is simple to implement and able to exploit collision-free subcarriers to decode packets despite successive collisions in overlapping channels. We apply Remap to 802.11g to demonstrate that the diversity created by remapped packets can substantially improve decoding efficiency and improve wireless throughput. We implement our technique in software radio and demonstrate that it has potential to be deployed with simple software and firmware updates.

1. INTRODUCTION

As OFDM becomes the foundation of modern high-speed wireless networks due to its advantages such as lower symbol rate, effective usage of a large frequency band, and resistance to frequency-selective fading, collisions in overlapping OFDM channels become an increasingly important problem in the deployment of high-speed wireless networks. Specifically, in an OFDM network, each channel is allocated a set of subcarriers, and two channels overlap when the intersection of their sets of subcarriers is not empty. Consider 802.11g, which is becoming almost ubiquitously deployed in many residential neighborhood. With only 3 orthogonal channels with disjoint sets of subcarriers but a large number of access points in densely populated neighborhood, it is inevitable that many 802.11 access points in range of each other use overlapping channels, as observed by previous measurement studies (e.g., [1]). In [15], the authors show that partially overlapping channels may improve network throughput even in managed 802.11 networks, when the number of orthogonal channels is limited. Channel overlapping is also allowed in WiFi networks built on digital white spaces [2]. Although one may try to alleviate the shortage of orthogonal channels by using variable bandwidth channels, as advocated in [5, 9], bursty or time varying workload can pose a problem for channel width adaptation.

However, collisions in overlapping OFDM channels dur-

ing contention and/or in the presence of hidden terminals are distinct from collisions in a single-channel setting. Consequently, recent progress (e.g., Zigzag Decoding [6]) on handling single-channel collisions does not directly apply.

In this paper, we present Remap, a simple, novel paradigm for handling collisions in OFDM networks with overlapping channels. Remap is different from the existing, passively repeat paradigm, and introduces a novel concept called retransmission permutation to permute the bit-to-subcarrier mapping after each retransmission. Retransmission permutation is a powerful diversity technique [17] that can recover frequency selective losses from subsequent retransmissions when there is no collision. When there are collisions, it in essence provides channel-width adaptation and allows bootstrapping of the decoding of collided packets that may otherwise be impossible to decode.

Specifically, the foundation of Remap is based on a simple observation and a simple idea. The observation is that when two packets transmitted in overlapping channels collide, only the subcarriers in the intersection of the two channels collide; the bits in other subcarriers are clean and can be collected. However, the non-colliding subcarriers do not contain complete packet information. The idea of Remap is to introduce structured permutation on the mapping from bits to subcarriers after each collision to create structured diversity. This diversity allows either independent decoding or bootstrapping other decoding techniques such as Zigzag decoding [6]. Integrating Remap with an existing system requires small changes to the OFDM physical layer as it involves only bit-to-subcarrier remapping.

In particular, we design 802.11g/Remap, which applies Remap to 802.11g to demonstrate its effectiveness. We show that by using the diversity created by remapped packets, an 802.11g receiver a can decode any packet P_a after 4 collisions with other transmissions in *adjacent* channels; the number of collisions reduces to 2 if the other transmissions are in *non-adjacent* channels. These numbers do not make any assumptions on the packets collided with P_a . If the packets collided with P_a are the same (i.e., both collisions are between P_a and another packet P_b), Remap can bootstrap decoding both packets with the bits on the collision-free subcarriers. Decoding both packets at a single AP is important so that the combiner (used in systems such as [16]) behind the AP can make use of the reception diversity. In contrast, without Remap, Zigzag cannot decode both packets. Furthermore, 802.11g/Remap is backward compatible with

*Bell Labs; †Microsoft Research Asia; ×Yale.

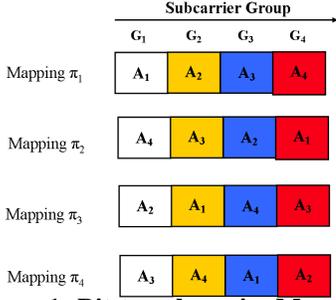


Figure 1: Bit-to-subcarrier Mapping.

802.11 MAC. Thus, 802.11g/Remap has potential to be deployed with simple software and firmware updates to the existing 802.11 networks.

We have implemented 802.11g/Remap using a software radio testbed. Our initial experimental results show that Remap works. The BER on decoding collision-free subcarriers is close to that of decoding collision-free transmissions, when the interference signal strength is smaller than the desired signal by at least 5 dB.

2. BASIC IDEA

802.11g Primer

We use 802.11g to illustrate our basic idea. In standard 802.11g, data bits are assigned to subcarriers. We denote the first group of 16 subcarrier frequencies of 802.11g as G_1 , the next group as G_2 , etc. Each 802.11g channel consists of 64 consecutive subcarrier frequencies¹. Thus, the first channel C_1 of 802.11g consists of four groups: G_1 , G_2 , G_3 , and G_4 ; channel C_2 consists of G_2 , G_3 , G_4 , and G_5 ; channel C_3 consists of G_3 , G_4 , G_5 , and G_6 ; channel C_4 consists of G_4 , G_5 , G_6 , and G_7 , etc. Note that C_2 is an *adjacent channel* of C_1 ; C_3 and C_4 are *overlapping non-adjacent channels* of C_1 ².

Assume that a sender uses channel C_1 to send a packet P consisting of data bits A_1 , A_2 , A_3 , and A_4 . Let the bit-to-subcarrier assignment be that $A_1 \rightarrow G_1$, $A_2 \rightarrow G_2$, $A_3 \rightarrow G_3$, and $A_4 \rightarrow G_4$. In other words, the bits A_1 are assigned to be carried by subcarrier group G_1 , A_2 by G_2 , A_3 by G_3 , and A_4 by G_4 . If the transmission for packet P is not successful, 802.11g retransmits the packet P where the bit-to-subcarrier assignment is the same.

Retransmission Permutation

The key novelty introduced by Remap is that during a retransmission, Remap uses a permutation scheduling of bit-to-subcarrier mapping.

Figure 1 is a schedule of bit-to-subcarrier mapping for 802.11g. As shown in the figure, for the original transmission, bits A_1, \dots, A_4 are mapped to subcarrier groups G_1, \dots, G_4 respectively. For the first retransmission, bits A_4, A_3, A_2, A_1 are mapped to subcarrier group G_1, \dots, G_4 respectively. The third and fourth rows are for the second and third retransmissions. We cycle through these four mappings for more

¹Only 48 subcarriers carry data bits

²Note that non-overlapping channels do not equal to orthogonal channels; due to imperfect filtering, guard bands are needed to achieve orthogonality. Our decoding technique is subject to adjacent channel interference.

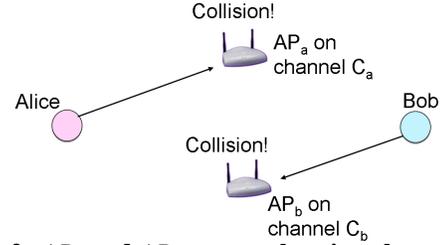


Figure 2: AP_a and AP_b use overlapping channels. Packet from Alice to AP_a and that of Bob to AP_b collide.

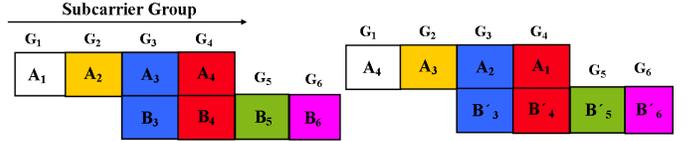


Figure 3: Subcarrier view of collisions: non-adjacent channels.

retransmissions. We only need 2 bits to encode the four mapping schemes. We can use the reserved bits in the SERVICE field of the PLCP header.

With the basic idea, now we demonstrate the benefits of Remap using a simple example. Consider two residential users, Alice and Bob, who use 802.11g to connect to access points AP_a and AP_b respectively. Let the channel between Alice and AP_a be C_a , and that between Bob and AP_b be C_b . Assume that the two channels are overlapping channels. Figure 2 shows the setting.

Due to hidden terminals or randomness, Alice may transmit a packet P_a to AP_a concurrently with Bob transmitting a packet P_b to AP_b , causing collisions at AP_a and AP_b . In absence of receiving an acknowledgment, Alice retransmits P_a , which may again collide with a transmission of Bob for packet P'_b . Note that P'_b may be different from P_b due to rescheduling. Without Remap, packet P_a cannot be decoded by the access point AP_a so long there are collisions.

Remap, however, allows decoding of collided packets. To illustrate our idea, we consider how AP_a decodes P_a . We will show that Remap allows AP_a to decode P_a after at most 3 retransmissions. We illustrate three cases.

Collisions in Non-adjacent Channels: Figure 3 considers the first case, when Alice and Bob use overlapping but non-adjacent channels (e.g., Alice uses C_1 and Bob uses C_3). The figure shows the collided packets in the frequency domain. We can check that AP_a can decode packet P_a after Alice retransmits the packet only once.

Collisions in Adjacent Channels: The worst-case of 3 retransmissions happens when Alice and Bob use adjacent channels (e.g., Alice uses C_1 and Bob uses C_2). Figure 4 shows the first transmission and one retransmission. After the two transmissions, bit blocks A_1 and A_4 will be recovered. Two

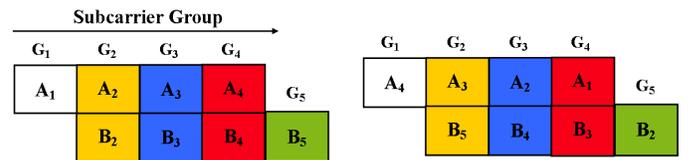


Figure 4: Subcarrier view of collisions: adjacent channels.

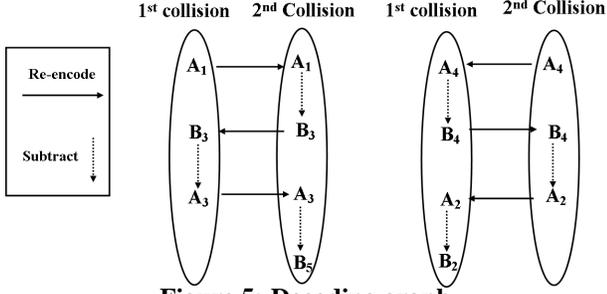


Figure 5: Decoding graph.

additional collisions will yield A_2 and A_3 , allowing full recovery of P_a .

Matching Collisions: The worst case of 3 retransmissions does not make any assumption on the packets that collided with P_a . When the first two packets that collided with P_a are the same, Remap can achieve higher efficiency. We refer to this case as *matching collisions*.

Specifically, for the case of matching collisions, Remap uses bits that are mapped to collision-free subcarriers to bootstrap more advanced decoding motivated by ZigZag decoding. Again see Figure 4. For the example, we have bit sequences A_1 and A_4 which are on collision-free subcarriers. A_1 is the bit sequence of all the symbols of P_a that are mapped to subcarrier group 1. We first decode them. Once we decode A_1 , we then re-encode them onto subcarrier group 4. We subtract the re-encoded signal from the second collision so that we can decode B_3 . When we get B_3 , we further re-encode them onto subcarrier group 3 and subtract from the first collision. After the subtraction, we can decode A_3 from the first collision. Similarly, A_3 causes B_5 to get decoded. We then decode from A_4 . A_4 causes B_4 to be decoded from the first collision. B_4 causes A_2 to get decoded from the second collision and followed by B_2 . Thus, we can decode all bits in all subcarriers, even the bits of the second packet transmitted in subcarriers outside of channel C_a in the first transmission! The decoding graph is shown in Figure 5. Note that, for simplicity, we have presented a simplified view of 802.11g subcarrier structure. For 802.11g, there are unused subcarriers, so we need to define subcarrier groups in terms of unused subcarrier groups. In this definition, the subcarrier groups are not aligned. However, this does not present any decoding problem. Interested readers can refer to a more complete version of this paper.

3. REMAP DETAILS

The preceding section presents the basic idea. In this section, we give more details on some key issues related with Remap. We use 802.11g to be concrete. Our presentation setting is that an 802.11g AP $_a$ tries to decode packets (P_a, P'_a) from Alice. We refer to the channel of Alice's transmissions as the primary channel. The channel for Bob's transmissions (say for packets P_b and P'_b) is called the secondary channel. We focus on the case of one secondary channel, but our scheme can be extended to multiple. Transmissions from the secondary channel cause interference to the primary channel. For lack of space, we omit channel estimation, frequency offset estimation and timing acquisition.

3.1 Detecting Collisions

To detect a collision, the AP exploits the known preamble with which every packet starts. The 802.11g preamble consists of 10 identical short symbols, each of 16 samples, and 2 identical long symbols, each of 64 samples. A 802.11g receiver does auto-correlation of the first 160 received samples. It then performs correlation to accurately locate the start point of the preamble. Because the preamble is independent of Alice and Bob's data, the correlation of the preamble samples with Alice and Bob's data should be close to zero.

The correlation technique can also be used to detect collision. If the correlation peaks in the middle of a packet, it indicates that there is likely a collision. We now need to determine on which channel the collided packet is transmitted. We use standard energy-detection technique. We illustrate using 802.11g. Each 802.11g channel consists of 64 subcarriers (a 20 MHz channel). As discussed before, we partition them into 4 groups. We try to find out which group's energy has a significant change before and after the correlation peak. Based on 802.11g's channel structure, once we determine the subcarrier groups that experience a collision, we can infer on which channel the second packet is transmitted.

3.2 Detecting Matching Collisions

Once an AP determines that the received signal is the result of a collision, it will search for a matching collision (a collision of the same two packets). Each AP stores recent unmatched collisions. Because senders in 802.11 keep retransmitting a failed transmission as soon as the medium is free, it is sufficient to store a few most recent collisions.

We again use correlation to match the current collision against stored collisions. Assume that the AP is trying to match two collisions (P_a, P_b) and (P'_a, P'_b). The AP knows the channels that each of these four packets are transmitted on and the timing offsets. Without loss of generality, let us assume that P_a and P'_a are transmitted on the primary channel. Because Remap maps bits to different subcarriers during retransmissions, we cannot simply perform a correlation in the time domain.

We first describe matching collision packets on the primary channel. We perform a FFT on the received mixed samples y of collision (P_a, P_b) and y' of collision (P'_a, P'_b) using the timing of Alice's packet. Let N be the number of subcarriers and M the number of symbols in each packet. After correcting frequency offsets, let $Y[l, i], Y'[l, i]$ be the i -th subcarrier sample of the l -th symbol in the frequency domain for y and y' respectively. Let $Y^*[l, i]$ denote the complex conjugate of $Y[l, i]$. Let $H_a[i]$ and $H'_a[i]$ be the channel gains associated with the i -th subcarrier of Alice's channel to her AP. Let $\hat{Y}'[l, i]$ be the reverse-mapped samples. That is, if subcarrier i is mapped to subcarrier k in P'_a , then $\hat{Y}'[l, i] = H'_a[i]Y'[l, k]/H_a[k]$. The correlation is computed as follows:

$$\begin{aligned} \Gamma(P_a, P'_a) &= \sum_{l=1}^M \sum_{i=1}^N Y^*[l, i] \hat{Y}'[l, i] \\ &= \sum_{l=1}^M \sum_{i=1}^N |Y_a[l, i]|^2. \end{aligned} \quad (1)$$

Note that, we assume that the channel of each subcarrier for the retransmitted packet does not change much from the original transmission. Note that, if AP_a does not know the bit-to-subcarrier mapping of P'_a , it may need to try out all four mappings.

For matching collision packets on the secondary channel, we cannot take all subcarrier groups into account. This is because subcarriers outside the primary channel are filtered at AP_a . Instead, we do the same correlation, but with a subset of subcarrier groups.

3.3 Detecting Modulation

Note that, given one collision, we cannot decode the PLCP header of the packet from the secondary channel. Thus, we do not know the modulation scheme that the second packet is encoded with. *Remap solves this problem nicely.* We assume that the channel has not changed much between the two collisions. This is reasonable given that 802.11 is used primarily in indoor environments. *During the Remap decoding process of adjacent channel collisions, rather than decoding and re-encoding P_b , we subtract the raw samples of P_b .* We omit the details of raw sample subtraction due to lack of space. The drawback from decode and re-encode subtraction is added noise. However, PLCP is redundantly encoded; so it is noise resilient. Once we obtain the PLCP header from the first pass of Remap, we can then apply the normal Remap decoding process. For subsequent bits of the second packet, we decode using the modulation scheme encoded in its PLCP header's signaling field.

3.4 Remap Decoding

The full Remap decoding algorithm is shown in Figure 6. *Remap_decode* takes a new collision and the set of saved collisions. *channel* is the channel that the receiver tunes to.

We first detect whether a collision happens on all 4 subcarrier groups. If so, then the collision is on the same channel. We pass it to the Zigzag decoder (modified to deal with bit-to-subcarrier remapping). If it is a collision on overlapping channels, we will run through the set of collisions *colset*. We first see if it is a matching collision pair; if so, we decode matching collisions. If it is not a matching collision pair, we will try to see if y' and y contain a matching packet in the primary channel *channel*. If that is the case, we will collect bits on collision-free subcarriers.

When we finish the loop, we will further collect collision-free subcarriers from the new collision y . We will then try to reconstruct the *pkt* in channel *channel*. If we have enough bits and the decoding passes CRC check, we will return *pkt*. Otherwise, we will return NULL.

We characterize the effectiveness of Remap for matching collisions using the following theorem:

THEOREM 1. *Given two matching collision pairs as shown in Figure 2. Alice uses bit-to-subcarrier mapping $\pi_{a,i}$ and $\pi_{a,i'}$ for the first and second transmissions on channel C_a respectively. Similarly Bob uses mapping $\pi_{b,j}$ and $\pi_{b,j'}$ on C_a 's adjacent channel C_b . The four mappings are chosen as*

```

Remap_decode(colset, y, channel) // colset is the set of collisions
01. scg = colli_subcarrier_groups(y)
02. if (|scg|==4) // co-channel collision
03. Zigzag_decode(colset, y) // account for bits-to-subcarrier remapping
04. else
05.   foreach  $y' \in colset$ 
06.     if matching_collisions(y, y', channel)
07.       bit_collector = Remap_decode_matching_pair(y, y', channel)
08.     else matching_packet_in_channel(y, y', channel)
09.       bit_collector=collect_subcarrier_group_bits(y', channel)
10.   endif
11. end foreach
12. endif
13. bit_collector=collect_subcarrier_group_bits(y, channel)
14. pkt=reconst_pkt(bit_collector)
15. colset = colset  $\cup$  y
16. return pkt

```

Figure 6: Remap decoding algorithm.

shown in Figure 1 on their respective channels. If $\pi_{a,i} \neq \pi_{a,i'}$ and $\pi_{b,j} \neq \pi_{b,j'}$, then Remap can decode both packets P_a and P_b at AP_a or AP_b .

Due to symmetry of the four mappings shown in Figure 1, w.l.o.g., we assume that the first pair uses mapping $\pi_{a,1}$ and $\pi_{b,1}$. Figure 4 shows one of the 9 cases. It can be decoded solely in the frequency domain iteratively. However, for three of the 9 cases, we have to resort to decoding in the time domain iteratively for a subset of subcarrier groups.

3.5 Issue with Scrambler

All of the bits transmitted by 802.11g are scrambled using a frame synchronous 127 bits sequence generator. Scrambling is used to randomize the input bits to prevent long strings of 1s and 0s. The initial state of the scrambler varies from packet to packet. To enable estimation of sender scrambler state, 802.11g sets the seven least significant bits of the SERVICE field of the PLCP header to all zeros. Scrambling is done after convolutional encoding.

To enable Remap decoding when the scrambler of the sender changes state from packet to packet, we need to descramble the received samples of a packet (with collisions). To descramble, we need to know the scrambling seed. For simplicity, we assume that Remap simply uses the same seed for all retransmissions.

3.6 Loss of Orthogonality

When decoding Alice's packet on the primary channel, loss of orthogonality can happen for various reasons: (1) frequency offset of the second packet; (2) if symbols are unaligned between Alice and Bob's packets, when performing an FFT, the energy of Bob's packet on subcarrier i may spread to Alice's subcarriers (not limited to subcarrier i), as Alice's FFT window will not contain complete OFDM symbols of Bob's packet; (3) energy leakage from adjacent non-overlapped subcarriers.

Because each 802.11g channel has 6 unused subcarriers at the beginning and the end, energy leakage affecting subcarriers beyond 6 will be very small. Thus, we only need to handle (1) and (2).

After Remap decoding decodes P_a and P_b with loss-of-

orthogonality interference, we can reconstruct it. We then subtract the interference to better decode P_a . This can be done similarly for P_b . In theory, this process can iterate. The quality of this iterative interference cancellation technique will depend on the decoding quality of the first step. We do not investigate these kinds of techniques in the experiments of this paper.

4. EVALUATIONS

4.1 Experimental Setup

We evaluate Remap using the Sora software radio platform [13]. Sora has a full-fledged 802.11g physical layer. The OFDM implementation uses the 802.11g standard preamble structure. We have implemented both Remap decoding of non-matching collisions and matching collisions. However, we focus on evaluating Remap decoding of the non-matching collision case. Remap can work with a variety of modulation schemes. In this evaluation, we use Binary Phase Shift Keying (BPSK). BPSK is used for the 6 Mbps data rate of 802.11g.

We perform the following experiments. We set up one Sora receiver, and two Sora senders Alice and Bob. We set Alice to transmit on channel 3. We fix the receiver at channel 3 as well. We change the locations of these nodes to get different SNRs. We perform two experiments for each location. In the first experiment, we set Bob to transmit at channel 4; in the second, we set Bob to transmit at channel 5. For each experiment, we let Alice and Bob send continuously for a given amount of time. We call the data collected a dump. We perform 100 independent dumps. Each dump lasts about 100 micro-seconds and contains 10 to 20 independent packets. For each packet from Alice, we send the original version and 3 additional Remap packets. Bob continuously sends randomly generated packets. After we collect the dump data, we post-process them. We first locate the preamble of each channel 3 packet and cut the samples into packets. We then group these packets into groups of 4 packets each. Each group of 4 packets ideally contains the original transmission of a packet and its three Remap packets. We adjust these 4 groups through our collision matching algorithm. We then decode each of the 4 packet samples. We collect BER and throughput. Since we know the content of Alice’s packets, we know whether each demodulated bit is correct or not. We then use the 802.11g Viterbi decoder to decode the packet. If the packet passes through the CRC check, we then count the packet as decoded correctly. Otherwise, we count it as an error packet. Note that, we can decode at most two packets from each 4-packet group for collisions with channel 5, and at most one packet for collisions with channel 4. We normalize the system throughput of each setting with the ideal throughput where all collisions can be decoded. We refer to it as the *normalized throughput*.

We conduct experiments at multiple locations inside a large office building. There are rarely any background 802.11b/g activities. We report results from two locations. At both locations, we also perform decoding of normal collision-free

transmissions of channel 3 packets. Almost all packets can be decoded. Furthermore, we decode the collision packets using the normal 802.11 decoder (the same decoding capability of a successive interference cancellation decoder). The throughput is close to zero (a few packets out of about 1500 packets).

4.2 Non-adjacent Channel Collision

Table 1 shows the overall throughput of two SNR settings. Note that the SNR of 21.2 dB of channel 5 is the measured SNR at the receiver who is tuned to channel 3. We see that, with SNR difference about 5 dB and 9 dB between channel 3 and channel 5 received signal strength, the normalized throughput is 0.81, and 0.98.

Scenario → SNR (ch3,ch5) ↓	Non-adjacent channel	
	BER	Throughput
(26.3dB, 21.2dB)	3.3×10^{-3}	0.81
(22.3dB, 13.5dB)	6.6×10^{-4}	0.98

Table 1: BER and normalized throughput at different locations.

To have a detailed understanding of the BER of each collision group and the throughput of each dump, we plot the scatter plot of each. We show only the second experiment. As shown in Figure 7, the BER of most collision pairs are very low. A few are very high around 25%. This is due to the fact that we missed the preamble of a channel 3 packet. So half of the subcarriers will be decoded incorrectly. With half randomly decoded correctly, this results in a BER of 25%. Since these are few, it shows that our preamble location (timing acquisition) algorithm performs well in these SNR settings.

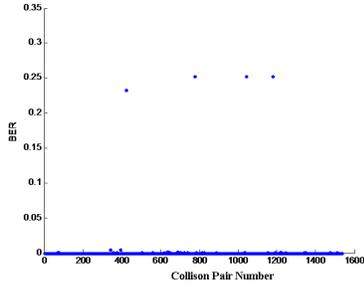
4.3 Adjacent Channel Collision

Now we look at the adjacent channel collision case. Table 2 shows the overall throughput of two SNR settings. Note that, in the second case when the SNR difference is 10dB the decoding probability is 0.91 and similar to the non-adjacent channel collision case (0.98). However, when the SNR difference is 3 dB, only half of the packets decoded can pass CRC check. When the SNR difference is 3 dB, the largest BER are clustered around 0.25 and 0.125. These are due to the error of our group separation algorithm (when interference is high). The rest of the packets that do not pass CRC are mostly due to high interference. The energy leaked from channel 4 causes the signal to interference and noise ratio to be low.

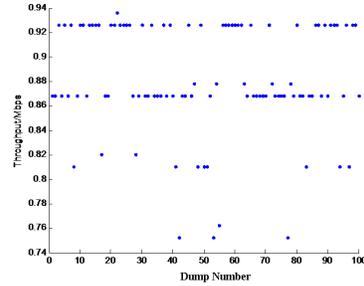
Scenario → SNR (ch3,ch5) ↓	Adjacent channel	
	BER	Throughput
(26.3dB, 23.4dB)	4.1×10^{-2}	0.50
(22dB, 12.0dB)	6.6×10^{-4}	0.91

Table 2: BER and normalized throughput at different locations.

5. RELATED WORK



(b) Scatter plot of BER



(d) Scatter plot of throughput

Figure 7: Experiments on non-adjacent channel collision (channels 3 and 5).

Remap’s key contribution is a novel bit-to-subcarrier remapping technique for resolving collisions in overlapping channels. It is different from prior techniques on interference cancellation and joint decoding (e.g., [19, 14, 10]). These techniques operate on a single collision, and their decoding capability is limited by the channel capacity. Remap is related to analog network coding [12]. However, a receiver capable of analog network coding can decode a collision only if it knows one of the collided packets.

Zigzag is designed in the context of no bit-to-subcarrier remapping. Remap builds on top of Zigzag. Remap works for non-overlapping channel collisions and is designed to take advantage of bit-to-subcarrier remapping. Unlike Zigzag, Remap does not rely on matching collisions. In the case of two matching collisions, Remap enables diversity combining from receptions at multiple APs as used in [16]. The reason is that Remap can decode both packets involved in the collisions as long as the number of overlapping subcarriers is no fewer than half of the total.

Remap differs from prior work on MAC layer techniques (e.g., [11, 3]) to avoid collisions. Remap decodes collisions and works with 802.11 MAC without modifications.

The concept of decoding interfering users has been an area of intense study in communication and information theory. Techniques such as MIMO [18] and interference alignment [8, 7, 4] require synchronization, coding and scheduling. In contrast, Remap resolves collisions without these requirements.

6. CONCLUSIONS AND FUTURE WORK

We presented Remap, a simple, novel paradigm that uses retransmission permutation for decoding collisions in overlapping OFDM channels. Our prototype implementation demonstrates that Remap can be simple to implement and able to exploit collision-free subcarriers to decode packets despite successive collisions.

For future work, we will deploy a larger testbed based on our Sora nodes, and understand how Remap works in diverse settings. We especially would like to investigate techniques that deal with the loss-of-orthogonality issue, and see how well matching-collision decoding works.

We believe that Remap can offer key insight in resolving collisions in other OFDM based wireless systems that have different subcarrier and channel structures. We are investi-

gating techniques inspired by Remap in the context of future dynamic spectrum access networks, where different nodes may select overlapping frequency bands.

7. REFERENCES

- [1] A. Akella, G. Judd, S. Seshan, and P. Steenkiste. Self-management in chaotic wireless deployments. In *Proceedings of MobiCom*, 2005.
- [2] P. Bahl, R. Chandra, T. Moscibroda, R. Murty, and M. Welsh. White space networking with Wi-Fi like connectivity. In *Proceedings of SIGCOMM*, 2009.
- [3] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. MACAW: A media access protocol for wireless LANs. In *Proceedings of ACM SIGCOMM*, 1994.
- [4] V. R. Cadambe and S. A. Jafar. Interference alignment and the degrees of freedom for the k user interference channel. Preprint, 2007.
- [5] R. Chandra, R. Mahajan, T. Moscibroda, R. Raghavendra, and P. Bahl. A case for adapting channel width in wireless networks. *SIGCOMM Comput. Commun. Rev.*, 38(4):135–146, 2008.
- [6] S. Gollakota and D. Katabi. Zigzag decoding: combating hidden terminals in wireless networks. *SIGCOMM Comput. Commun. Rev.*, 38(4), 2008.
- [7] S. Gollakota, S. D. Perli, and D. Katabi. Overcoming the antennas-per-node throughput limit in MIMO LANs. Technical Report MIT-CSAIL-TR-2009-007, MIT, 2009.
- [8] K. S. Gomadam, V. R. Cadambe, and S. A. Jafar. Approaching the capacity of wireless networks through distributed interference alignment. In *Proceedings of IEEE GLOBECOM*, 2008.
- [9] R. Gummadi, R. Patra, H. Balakrishnan, and E. Brewer. Interference Avoidance and Control. In *Proceedings of Hotnets-VII*, 2008.
- [10] D. Halperin, T. Anderson, and D. Wetherall. Taking the sting out of carrier sense: interference cancellation for wireless LANs. In *Proceedings of MobiCom*, 2008.
- [11] P. Karn. MACA - a new channel access method for packet radio. In *ARRL/CRRLL Amateur Radio 9th Computer Networking Conference*, Sept. 1990.
- [12] S. Katti, S. Gollakota, and D. Katabi. Embracing wireless interference: Analog network coding. In *Proceedings of ACM SIGCOMM*, 2007.
- [13] Kun Tan et al. Sora: High performance software radio using general purpose multi-core processors. In *Proceedings of NSDI*, 2009.
- [14] L. E. Li, R. Alimi, R. Ramjee, J. Shi, Y. Sun, H. Viswanathan, and Y. R. Yang. Superposition coding for wireless mesh networks. In *Proceedings of MobiCom*, 2007.
- [15] A. Mishra, V. Shrivastava, S. Banerjee, and W. Arbaugh. Partially overlapped channels not considered harmful. *SIGMETRICS Perform. Eval. Rev.*, 34(1):63–74, 2006.
- [16] A. Miu, H. Balakrishnan, and C. E. Koksal. Improving loss resilience with multi-radio diversity in wireless networks. In *Proceedings of MobiCom*, 2005.
- [17] H. Samra and Z. Ding. Retransmission diversity schemes for multicarrier modulations. *Wireless Communications, IEEE Transactions on*, 5(5):1142–1147, 2006.
- [18] D. Tse and P. Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, May 2005.
- [19] S. Verdú. *Multisuser Detection*. Cambridge University Press, New York, NY, 1998.