Partial OFDM Symbol Recovery to Improve Interfering Wireless Networks Operation in Collision Environments

Waseem Ozan, *Member, IEEE*, Izzat Darwazeh, *Senior Member, IEEE*, and Kyle Jamieson, *Senior Member, IEEE, ACM*

***Abstract*— The uplink data rate region for interfering transmissions in wireless networks has been characterised and proven, yet its underlying model assumes a complete temporal overlap. Practical unplanned networks, however, adopt packetized transmissions and eschew tight inter-network coordination, resulting in packet collisions that often partially overlap, but rarely ever completely overlap. In this work, we report a new design called Partial Symbol Recovery (PSR), that specifically targets the parts of data symbols that experience no interference during a packet collision. PSR bootstraps a successive interference cancellation (SIC) like decoder from these strong signals, thus improving performance over techniques oblivious to such partial packet overlaps. We have implemented PSR on the WARP software-defined radio platform and in trace-based simulation. Our performance evaluation presents experimental results from this implementation operating in a 12-node software network testbed, spread over two rooms in a non-line-of-sight indoor office environment. Experimental results confirm that our proposal PSR decoder is capable of decoding up to 60% of collided frames depending on the type of data and modulation used. This consistently leads to throughput enhancement over conventional Wi-Fi under different scenarios and for the various data types tested, namely downlink bulk TCP, downlink video-on-demand, and uplink UDP.**

***Index Terms*— Partial symbol recovery, wireless communications, 802.11, OFDM, collision environments, hidden terminals, interference cancellation.**

I. Introduction

IN DENSELY populated urban areas, wireless networks that exist in unlicensed spectrum (such as 802.11) routinely contend with vast numbers of users operating nearby, causing uncontrolled amounts of interference, with transmissions overlapping in both time and frequency. While medium access protocols can avoid some collisions, packet collisions due to hidden terminal nodes and inter-network collisions are still prevalent and harm throughput [1], [2]. Roughly speaking, the likelihood of a bit error is positively correlated with the ratio of signal power to noise plus interference power (SINR), measured at the receiver. The result is that in such unplanned networks, problems of interference from neighboring networks combine with the problems of range and reliability these networks already experience, resulting in degradation of application performance.

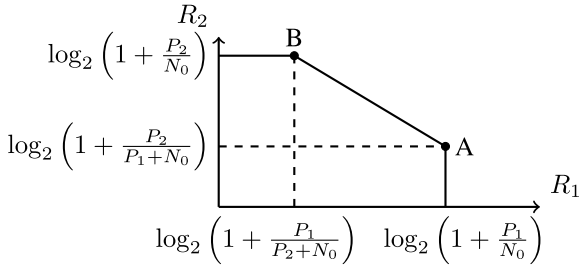


Fig. 1. Capacity region of the two-user uplink Gaussian noise interference channel [3].

From an information-theoretic standpoint, Ahlswede [4] and Liao [5] have characterised the *capacity region* of the general multiple access channel, *i.e.*, the set of tuples (each entry in the tuple containing the rate of each user) such that there exists a receiver design that achieves an arbitrarily low bit error rate provided that the users transmit at rates within that set. Fig. 1 shows the capacity region for the two-user uplink Gaussian noise channel [3]. This line of work proved that in such a channel, the rate of each user cannot exceed the Shannon capacity of interference-free links (leading to the constraints that $R_{1,2} < \log(1 + P_{1,2}/N_0)$, where $P_{1,2}$ are the transmit powers of users one and two respectively, and N_0 is the Gaussian noise power) and that the total rate cannot exceed the Shannon capacity of a Gaussian noise channel with power equal to the sum of the two users' powers. The receiver that achieves Pareto-optimal points in the capacity region is the *successive interference cancellation* (SIC) receiver [3], which achieves point A or B in the capacity region by first treating one transmission as noise while decoding the

Manuscript received 21 June 2020; revised 12 September 2021 and 22 March 2022; accepted 1 August 2022; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor D. Hay. The work of Waseem Ozan was supported by the Electronic and Electrical Engineering Department, University College London, during his visit in Princeton University. *(Corresponding author: Waseem Ozan.)*

Waseem Ozan was with the Department of Electronic and Electrical Engineering, UCL, London WC1E 7JE, U.K., and also with the Department of Computer Science, Princeton University, Princeton, NJ 08540 USA. He is now with MediaTek Wireless, Cambridge CB23 6DW, U.K. (e-mail: waseem.ozan@mediatek.com).

Izzat Darwazeh is with the Department of Electronic and Electrical Engineering, UCL, London WC1E 7JE, U.K. (e-mail: i.darwazeh@ucl.ac.uk).

Kyle Jamieson is with the Department of Computer Science (and associated with the Department of Electrical and Computer Engineering), Princeton University, Princeton, NJ 08540 USA (e-mail: kylej@princeton.edu).

Digital Object Identifier 10.1109/TNET.2022.3202857

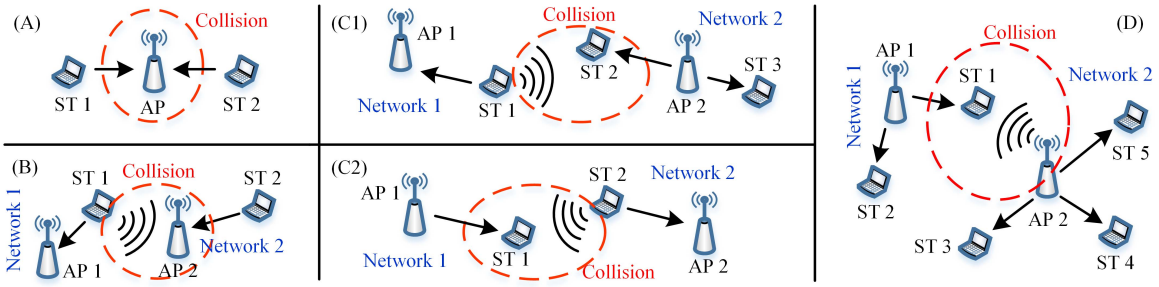


Fig. 2. PSR applies to different collision scenario: (A) is a hidden terminal scenario, which causes collisions within a network; (B, C1, C2 and D) depict inter-network collision scenarios that occur between two existing neighbouring networks. Nodes surrounded with dashed lines experience collisions.

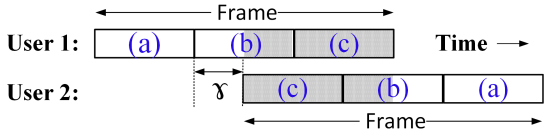


Fig. 3. Stations 1 and 2 transmit concurrently, leading to a collision. Different data symbols (a)–(c) experience differing degrees of overlap.

other transmission, then re-modulating and subtracting the first transmission from the received signal.

For years, work in this area took the above model—which treats every bit in two interfering transmissions equally—as the best fit to the reality of interfering data transmissions in wireless networks. Thus, this concept of capacity was treated as a given for single antenna communication, and efforts instead focused on other promising directions to increase capacity (such as the use of multiple antennas), which we survey in Section II.

However, in practical unplanned wireless networks, such as 802.11 there are many opportunities for collisions both within networks, as shown in Fig. 2. Our insight begins with the observation that in a practical unplanned wireless network such as 802.11, or uncoordinated mobile cellular networks, the interference channel model is not necessarily exhaustively applicable to every bit in the colliding packets, as Fig. 3 illustrates. From two colliding packet transmissions, we note **(a)** a head of some data symbols "in the clear" of the collision, **(b)** two partially-overlapping symbols of each colliding packet, and finally **(c)** the remainder of the wholly-overlapping symbols in the body of the packet overlap. Crucially, out of these three regions of packet data involved in the collision, only region **(c)** and the collided part of region **(b)** are subject to the Ahlswede-Liao multi-user capacity region. Therefore, there is an opportunity to tailor a receiver to operate beyond the rate region by utilising the collision-free part of region **(b)** to fully recover region **(b)**, then exploiting that to decode and recover region **(c)**. Achieving this, however, requires that two challenges be overcome:

1) First, for the symbols in region **(b)** that are partially overlapped by interference, a decode attempt on just the interference-free part of these symbols results in interference across different orthogonal frequency division multiplexing (OFDM) subcarriers in frequency. While the Sphere decoder [6], [7] can take this interference into account, however, its computational complexity increases exponentially with the number of subcarriers and so to make partial symbol recovery practical, we need to develop a computationally-efficient receiver whose performance approaches that of the Sphere decoder.
2) Second, channel coding is employed with the symbol-by-symbol modulation shown in Fig. 3, and this channel coding spreads information across the symbols in time, blurring the boundaries between the three regions noted above. Thus any receiver that leverages our observation for increased capacity must be co-designed with the channel coding in use.

This paper presents a new technique termed *Partial Symbol Recovery* (PSR), a physical- and data link-layer wireless receiver design that explicitly treats non- and partially-overlapping OFDM symbols differently from completely-overlapping OFDM symbols involved in a collision. The PSR algorithm first decodes the non-overlapping part of one user's overlapping OFDM symbol sent from one of the users, using a novel *interference modeling* step that models and cancels the intercarrier interference that results from truncation of the OFDM symbol in time. Our design also applies the Sphere decoder to small blocks of OFDM subcarriers in order to precisely cancel this intercarrier interference, while relying on linear methods to cancel interference across blocks, thus bounding the computational complexity entailed by the Sphere decoder. Furthermore, our design jointly integrates the functions of intercarrier interference cancellation, channel equalisation, carrier frequency offset (CFO) and phase offset compensation, and inter-symbol interference (ISI) mitigation, co-designing this functionality with the remainder of the system. This functionality comprises the *inner-loop decoder*, which is described fully in Section III-B.1.

After using the inner-loop decoder to make an estimate of the interference-free portion of an OFDM symbol, PSR re-modulates the resulting bits to a full-length OFDM symbol and cancels the other user's interference from the originally received signal, allowing it to make a decoding attempt on a complete interference-free symbol, provided the partially-received symbol was decoded correctly: this allows for a *window* of two symbols to be decoded jointly, and another symbol from the first user to be rendered interference-free. This process continues, expanding the window until the number

of bit errors falls to an error floor (determined by power and noise levels, and modulation/coding rates) that are determined empirically. The window then slides along the entire length of the collision, comprising the *outer-loop decoder* of the PSR decoding algorithm.

Our PSR performance results, presented in Section V, reveal and confirm the following findings:

- **Two-user rate region**: PSR operates with a larger two-user rate region compared to the one achieved using the SIC decoder;
- **Decoding capability**: experimentally proved that PSR decoder is capable of decoding on average 59% and 25% of collided frames, for BPSK and QPSK modulation formats, respectively;
- **Throughput performance**: results confirm an average throughput improvement over 802.11n of 10–21% for downlink bulk TCP, depending on modulation used;
- **Retransmission performance**: PSR reduces the average rate of retransmissions by 63–81%, 57–81% and 28–49% in the above scenarios, respectively.

The rest of this paper is structured as follows. We outline related work in Section II, placing our contributions in further context of the many ideas for coping with interference in wireless networks. We detail the design of the PSR link and physical layers in Section III. Our testbed implementation is given in Section IV. PSR system performance is extensively evaluated in Section V. At last, the conclusion is given in Section VI.

II. Related Work

Faster than Nyquist (FTN) Modulation Most data transmission is linear: the result of summing a sequence of pulses. These pulses are almost always orthogonal (*i.e.*, invisible) to each other, meaning that if the current symbol is viewed at a symbol duration later in time, it is zero—this is the *Nyquist criterion* [8]. If the criterion is not met, the pulses interfere, but in 1975 Mazo showed that for a simple linear modulation the data pulses could in fact be compressed in time by a multiplicative factor of 0.802 [9], a speedup of approximately 25%, without a loss in performance. Follow-on work has generalised this concept to OFDM modulation, compressing the OFDM signal in frequency [10], [11] and both time and frequency [12]. Partial symbol recovery's inner-loop decoder can be viewed as a type of FTN modulation that differs from previous work because of its time truncation. Partial Symbol Recovery leverages a refined version of the truncated-OFDM (T-OFDM) decoder [13] for its inner-loop decoder, but builds atop T-OFDM to develop the outer decoder in its entirety.

Massive MIMO Base Stations [14], [15] leverage large numbers of antennas to increase spectral efficiency. In multi-user communication networks, a coordination scheme utilising machine learning is applied to minimize the collisions [16]. Interference alignment [17] (IA) techniques use multiple antennas to align different interfering transmissions, and have been realised in both wireless LAN [18] and cellular [19] settings. A practical three-stage hybrid analog-digital precoding architecture, occupying fewer RF chains, is proposed aiming for a nonorthogonal Internet of Things (IoT) signal in low-cost multiuser MIMO systems [20]. For the physical layer cross-technology communication (CTC), the work of ZigFi [21] leverages the channel state information of the received frames to enable WiFi receivers to decode data from ZigBee senders.

Bit Rate Adaptation based on acknowledgements [22] and SINR [23] and rateless codes [24], [25] attempts to maximise utilisation of individual links, but still does not exploit the partial overlap concept. Other influential link-centric approaches include full-duplex wireless links [26], [27], which are also independent of PSR contribution.

A prior work on exploiting packet overlaps in partial packet recovery (PPR) [2], aims to recover the collision-free part of colliding packets but does not attempt to recover their colliding parts, hence it requires retransmission and suffers reduced throughput. Other work for decoding collisions from hidden terminal nodes, termed Zigzag Decoding [1], requires two successive collisions of the same packets. In contrast, partial symbol recovery is the first work we are aware of, besides the classical SIC algorithm (against which we benchmark in Section V), that recovers full packets from a single collision and requires no retransmission to operate.

III. Design

PSR is implemented jointly in the physical and data link layers. In this section, the mathematical derivation for decoding a partial overlapped symbol is given in Section III-A; the physical layer (L1) design of PSR is discussed in Section III-B; the data link layer (L2) design is described in Section III-C, and Section III-D addresses experimental design considerations for different parameters.

A. Mathematical Representation

1) At the Transmitter: The 802.11a,g signals consist of a stream of OFDM symbols, where each symbol has $Q = 64$ carriers, each of which carries $N = 48$ complex data symbols, 4 pilot subcarriers and the rest are padded zeros. Hence, the symbols before the modulation process (i.e. before the complex symbols are fed to the input of the IDFT process) are given by $\mathbf{d} = [d_1, \ldots, d_Q]^T \in \mathbb{C}^{Q\times 1}$. The discrete OFDM symbol is given in matrix form as:

$$\mathbf{x} = \mathbf{\Phi}\mathbf{d} \tag{1}$$

where $\mathbf{x} \in \mathbb{C}^{Q\times 1}$ is a vector of time samples representing one OFDM symbol; $\mathbf{\Phi} \in \mathbb{C}^{Q\times Q}$ is the sampled IDFT matrix and $\mathbf{d} \in \mathbb{C}^{Q\times 1}$ is the vector of input data symbols.

Using the conventional CP technique, the last $\mu \in \mathbb{N}$ samples of a symbol, forming the CP part $\mathbf{x}_{cp} \in \mathbb{C}^{\mu\times 1}$, are added to the beginning of each transmitted symbol giving:

$$\mathbf{s} = \begin{bmatrix} \mathbf{x}_{cp} \\ \mathbf{x} \end{bmatrix}_{V\times 1} \tag{2}$$

where $\mathbf{s} \in \mathbb{C}^{V\times 1}$ is the CP-OFDM symbol, where $V = Q+\mu$. A stream of OFDM symbols for user 1 are given by $\mathbf{s}_{user1}$.

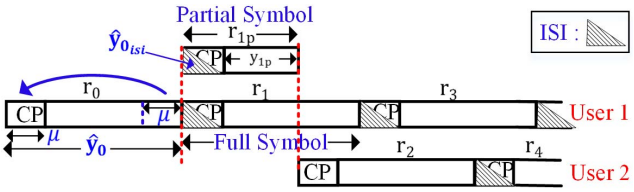


Fig. 4. A partial and full symbol in received colliding frames. The partial symbol, $\mathbf{r}_{1_p}$, consists of a CP and a part of a symbol. Symbols $\mathbf{r}_0, \mathbf{r}_1, \mathbf{r}_3$ belong to user 1 and the symbols $\mathbf{r}_2, \mathbf{r}_4$ belong to user 2. Each CP is contaminated with ISI.

2) At the Receiver: The signal is transmitted through a wireless frequency selective channel with channel impulse response (CIR) $\mathbf{h} = [h_0, h_1, \ldots, h_\mu] \in \mathbb{C}^{(\mu+1)\times 1}$. The mathematical representation of the received user 1 signal after traversing a multipath channel is given in a linear convolution process of the signal with the CIR as:

$$\mathbf{r}_{x-signal} = \begin{bmatrix} \vdots \\ \mathbf{y}_{-1_{isi}} + \mathbf{y}_{0_{cp}} \\ \mathbf{y}_0 \\ \mathbf{y}_{0_{isi}} + \mathbf{y}_{1_{cp}} \\ \mathbf{y}_1 = \begin{bmatrix} \mathbf{y}_{1_p} \\ \mathbf{y}_{1_r} \end{bmatrix} \\ \mathbf{y}_{1_{isi}} + \mathbf{y}_{x_{cp}} \\ \vdots \end{bmatrix} = \mathbf{h} * \mathbf{s}_{user1} \tag{3}$$

where the components appear in the column are: the first line $\mathbf{y}_{-1_{isi}} + \mathbf{y}_{0_{cp}} \in \mathbb{C}^{\mu\times 1}$ is the CP part of symbol $\mathbf{s}_0$ that is affected by the inter-symbol interference (ISI) components stemming from the previous symbol; $\mathbf{y}_0 \in \mathbb{C}^{Q\times 1}$ is the received symbol $\mathbf{s}_0$ after traversing the channel and the rest of the lines are for the next symbols. The symbol $\mathbf{y}_1$ is the symbol that encounter interference from the other user (i.e. user 2) in its second part ($\mathbf{y}_{1_r}$) and hence it is depicted as two parts collision-free part ($\mathbf{y}_{1_p} \in \mathbb{C}^{(\gamma Q-\mu)\times 1}$) and interfered part ($\mathbf{y}_{1_r} \in \mathbb{C}^{(V-\gamma Q)\times 1}$).

3) Utilising the CP Part: Out of the received signal given in (3), the decoder will act on the following parts of symbol 1, $\mathbf{r}_{1_p} \in \mathbb{C}^{\gamma Q\times 1}$ where $\gamma \leq 1$:

$$\mathbf{r}_{1_p} = \begin{bmatrix} \begin{bmatrix} \mathbf{y}_{0_{isi}} + \mathbf{y}_{1_{cp}} \end{bmatrix}_{\mu\times 1} \\ \mathbf{y}_{1_p} \end{bmatrix}_{\gamma Q\times 1} \tag{4}$$

where each collision-free partial symbol consists of a CP part and a fractional part of the OFDM symbol, which are the first and second lines in equation (4), respectively, this is depicted in Fig. 4. As the rest of the symbol is overlapped by the other user, then we use the CP in the decoding process to compensate for the portion lost due to the collision. This results in increase in the number of time samples in the partial symbol by the number of samples in the CP, and an increase in the energy contained in the partial symbol, thus resulting in reduced error rate. However, the CP part $\mathbf{y}_{1_{cp}}$ is distorted with ISI from the previous symbol $\mathbf{y}_{0_{isi}}$, hence, we estimate the ISI and subtract it from the partial symbol. This is achieved by convolving the channel impulse response (CIR, $\mathbf{h} \in \mathbb{C}^{(\mu+1)\times 1}$), with the estimated preceding symbol, $\hat{\mathbf{s}}_\mathbf{0}$, as:

$$\mathbf{h} * \hat{\mathbf{x}}_0 = \begin{cases} \hat{\mathbf{y}}_0, & 0 \leq k \leq Q-1 \\ \hat{\mathbf{y}}_{0_{isi}}, & Q \leq k \leq Q+\mu-1 \end{cases} \tag{5}$$

where $(*)$ is the convolution process. The output of the convolution comprises of $\hat{\mathbf{y}}_0 \in \mathbb{C}^{Q\times 1}$ and $\hat{\mathbf{y}}_{0_{isi}} \in \mathbb{C}^{\mu\times 1}$, which are the preceding symbol after traversing the multi-path channel, and the ISI component, respectively, as shown in Fig. 4. The interference-free CP may then be obtained by subtraction of the ISI ($\hat{\mathbf{y}}_{0_{isi}}$) from the partial symbol, $\mathbf{r}_{1_p}$, and the resulting symbol is given by:

$$\tilde{\mathbf{r}}_{1_p} = \mathbf{r}_{1_p} - \begin{bmatrix} \hat{\mathbf{y}}_{0_{isi}} \\ \mathbf{0}_{(\gamma V-\mu)\times 1} \end{bmatrix}_{\gamma Q\times 1} = \begin{bmatrix} \tilde{\mathbf{y}}_{1_{cp}} \\ \mathbf{y}_{1_p} \end{bmatrix}_{\gamma Q\times 1} \tag{6}$$

Next, we extend the partial symbol with zeros to length Q then we copy the first μ samples (i.e. the CP part, $\tilde{\mathbf{y}}_{1_{cp}}$) to the end of the zero-padded symbol. The signal can be represented by the channel matrix and the input signal as:

$$\begin{bmatrix} \tilde{\mathbf{y}}_{1_{cp}} \\ \mathbf{y}_{1_p} \\ \mathbf{0}_{(Q-\gamma Q)\times 1} \\ \tilde{\mathbf{y}}_{1_{cp}} \end{bmatrix} = \left[\begin{array}{c|c|c} \mathbf{H_a} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{H_b} \end{array}\right]_{V\times V} \begin{bmatrix} \mathbf{x}_{1_{cp}} \\ \mathbf{x}_{1_p} \\ \mathbf{0}_{(Q-\gamma Q)\times 1} \\ \mathbf{x}_{1_{cp}} \end{bmatrix}_{V\times 1} \tag{7}$$

where $\mathbf{0}$ are matrices of zeros and the matrices $\mathbf{H_a} \in \mathbb{C}^{\gamma Q\times \gamma Q}$ and $\mathbf{H_b} \in \mathbb{C}^{\mu\times\mu}$ are given as:

$$\mathbf{H_a} = \begin{bmatrix} h_0 & & & & \\ \vdots & \ddots & & & \\ h_\mu & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & h_\mu & \ldots & h_0 \end{bmatrix}_{\gamma Q\times\gamma Q} \tag{8}$$

$$\mathbf{H_b} = \begin{bmatrix} h_0 & \\ \vdots & \ddots \\ h_{\mu-1} & \ldots\ h_0 \end{bmatrix}_{\mu\times\mu} \tag{9}$$

where the empty space in the above matrix is for zero values.

After that, we remove the CP from the beginning of the symbol. Then, the matrix representation of the partial symbol is restructured so that the inter-samples components coming from the CP part are considered. Nevertheless, the channel matrix representation in the following equation is equivalent to that in (7). The partial symbol is represented by:

$$\bar{\mathbf{r}}_{1_p} = \begin{bmatrix} \mathbf{y}_{1_p} \\ \mathbf{0} \\ \tilde{\mathbf{y}}_{1_{cp}} \end{bmatrix}_{Q\times 1} = \left[\begin{array}{c|c|c} \mathbf{H_b} & \mathbf{0} & \mathbf{H_c} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{H_b} \end{array}\right]_{Q\times Q} \begin{bmatrix} \mathbf{x}_{1_p} \\ \mathbf{0} \\ \mathbf{x}_{1_{cp}} \end{bmatrix}_{Q\times 1} \tag{10}$$

where $\mathbf{0}$ are matrices of zeros and $\mathbf{H_c} \in \mathbb{C}^{\mu\times\mu}$ is given by:

$$\mathbf{H_c} = \begin{bmatrix} h_\mu & \ldots & h_1 \\ & \ddots & \vdots \\ & & h_\mu \end{bmatrix}_{\mu\times\mu} \tag{11}$$

From equation (10), we can observe two things: (i) the channel matrix is not a circulant matrix as it has missing elements in the middle of the channel matrix; and (ii) there is still missing samples in the partial symbol, which causes

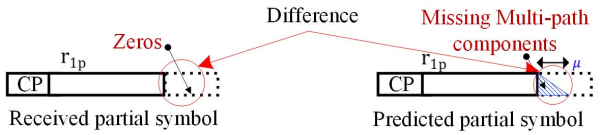


Fig. 5. A truncated partial symbol lacks certain multi-path components. The triangle depicts these estimated, missing multi-path components.

inter-carrier interference (ICI). Therefore, in the next steps we will show how to perform channel equalisation, find the ICI (covariance) matrix and how to decode the partial symbol using the covariance matrix.

4) Channel Equalisation: As mentioned earlier that the channel matrix is not circulant due to the missing multipath components in the partial symbol. To illustrate, we compare in Fig. 5 a received partial symbol with what such symbol would "hypothetically" be after traversing a multi-path wireless channel. The missing multi-path components, in received partial symbols, result in increased signal distortion level and consequently inaccurate one-tap equalisation process. Therefore, we derive the formula for the distortion coming from the missing multipath components so we can remove it in the decoder. Thus, we decompose the channel matrix in (10) into two matrices, one is the circulant matrix and the other is the missing elements matrix. Equation (10) will be re-written as:

$$\bar{\mathbf{r}}_{1_p} = \mathbf{H}\begin{bmatrix}\mathbf{x}_{1_p}\\ \mathbf{0}\\ \mathbf{x}_{1_{cp}}\end{bmatrix}_{Q\times 1} - \mathbf{H}_1\begin{bmatrix}\mathbf{x}_{1_p}\\ \mathbf{0}\\ \mathbf{x}_{1_{cp}}\end{bmatrix}_{Q\times 1} \tag{12}$$

where $\mathbf{H} \in \mathbb{C}^{Q\times Q}$ is the circulant channel matrix while $\mathbf{H}_1 \in \mathbb{C}^{Q\times Q}$ is the missing element channel matrix, which are the missing elements in equation (10). The second term ($\mathbf{r}_{dist}$) in the above equation is the distortion coming from the missing channel matrix elements (missing multipath components), hence this part should be estimated and subtracted from the partial symbol. Given that the symbol is not known at the receiver therefore the decoder is an iterative decoder, where in the first iteration the symbol will be estimated then used to generate the missing multipath components from the partial symbol during the next iterations. Thus, equation (12) after removing the distortion can be represented by:

$$\tilde{\bar{\mathbf{r}}}_{1_p} = \bar{\mathbf{r}}_{1_p} + \hat{\mathbf{H}}_1\begin{bmatrix}\hat{\mathbf{x}}_{1_p}\\ \mathbf{0}\\ \hat{\mathbf{x}}_{1_{cp}}\end{bmatrix}_{Q\times 1} = \mathbf{H}\begin{bmatrix}\mathbf{x}_{1_p}\\ \mathbf{0}\\ \mathbf{x}_{1_{cp}}\end{bmatrix}_{Q\times 1} \tag{13}$$

for the purpose of simplified mathematical representation we donnot show the impairments due to symbol estimation and hence the estimated symbol is supposed equal to the actual transmitted symbol (i.e. $\hat{\mathbf{x}}_{1_p} = \mathbf{x}_{1_p}$ and $\hat{\mathbf{x}}_{1_{cp}} = \mathbf{x}_{1_{cp}}$).

In addition, from [28] the eigenvalue decompostion of the circulant matrix can be given by $\mathbf{H} = \mathbf{M}\mathbf{\Lambda}\mathbf{M}^H$, where $\mathbf{\Lambda} = \text{diag}\{\lambda_0, \ldots, \lambda_{Q-1}\} \in \mathbb{C}^{Q\times Q}$ and λ_i is the ith eigenvalue of $\mathbf{H}$, while $\mathbf{M}$ and $\mathbf{M}^H \in \mathbb{C}^{Q\times Q}$ are unitary matrices, where $\mathbf{M}^H$ has rows that are the eigenvectors of $\mathbf{H}$ and $[.]^H$ denotes the conjugate transpose operation. Furthermore, from [28], it is shown that $\mathbf{M}^H$ is a discrete Fourier transform (DFT) matrix (i.e. $\mathbf{M}^H = \mathbf{\Phi}^H$ and $\mathbf{M} = \mathbf{\Phi}$). Therefore, the partial symbol $\tilde{\bar{\mathbf{r}}}_p$ after being passed to the DFT process is given by:

$$\begin{aligned}\bar{\mathbf{r}}_{1_{p-dft}} &= \mathbf{\Phi}^H\tilde{\bar{\mathbf{r}}}_{1_p} = \mathbf{\Phi}^H\mathbf{H}\begin{bmatrix}\mathbf{x}_{1_p}\\ \mathbf{0}\\ \mathbf{x}_{1_{cp}}\end{bmatrix} = \mathbf{\Phi}^H\mathbf{M}\mathbf{\Lambda}\mathbf{M}^H\begin{bmatrix}\mathbf{x}_{1_p}\\ \mathbf{0}\\ \mathbf{x}_{1_{cp}}\end{bmatrix}\\ &= \mathbf{\Phi}^H\mathbf{\Phi}\mathbf{\Lambda}\mathbf{\Phi}^H\begin{bmatrix}\mathbf{x}_{1_p}\\ \mathbf{0}\\ \mathbf{x}_{1_{cp}}\end{bmatrix} = \mathbf{\Lambda}\mathbf{\Phi}^H\begin{bmatrix}\mathbf{x}_{1_p}\\ \mathbf{0}\\ \mathbf{x}_{1_{cp}}\end{bmatrix}\\ &= \mathbf{\Lambda}\mathbf{\Gamma}\mathbf{d} = \boldsymbol{\lambda}\odot\tilde{\mathbf{d}}\end{aligned} \tag{14}$$

where $\mathbf{\Lambda}$ is a diagonal matrix, whose diaognal elements are the subchannels gain of all subcarriers $\boldsymbol{\lambda} \in \mathbb{C}^{Q\times 1}$, $\tilde{\mathbf{d}} = \mathbf{\Gamma}\mathbf{d}$ is the demodulated partial symbol when no multipath or noise channel present, $\mathbf{\Gamma}$ is the covariance matrix which is derived in the next section, and the notation ($\odot$) is the element-wise multiplication.

5) Inter-Carrier Interference: From equation (14), the transmitted time samples that correspond to the received partial symbol can be written as:

$$\begin{bmatrix}\mathbf{x}_{1_p}\\ \mathbf{0}\\ \mathbf{x}_{1_{cp}}\end{bmatrix} = \begin{bmatrix}\mathbf{x}_{1_p}\\ \mathbf{x}_{1_r}\\ \mathbf{x}_{1_{cp}}\end{bmatrix} - \begin{bmatrix}\mathbf{0}\\ \mathbf{x}_{1_r}\\ \mathbf{0}\end{bmatrix} = \mathbf{x} - \begin{bmatrix}\mathbf{0}\\ \mathbf{x}_{1_r}\\ \mathbf{0}\end{bmatrix} = \mathbf{\Phi}\mathbf{d} - \mathbf{\Phi}_p\mathbf{d} \tag{15}$$

where the second term ($\mathbf{\Phi}_p\mathbf{d}$) is the missing part of OFDM symbol at the receiver and hence $\mathbf{\Phi}_p$ is given by:

$$\mathbf{\Phi}_{p(k,n)} = \begin{cases}\mathbf{\Phi}_{(k,n)}, & \gamma Q - \mu \le k < Q - \mu\\ 0, & 0 \le k < \gamma Q - \mu\\ & \text{and} \quad Q - \mu \le k < Q\end{cases} \tag{16}$$

where $0 \le n < Q - 1$.

The partial symbol after the DFT process is given by:

$$\mathbf{\Phi}^H\begin{bmatrix}\mathbf{x}_{1_p}\\ \mathbf{0}\\ \mathbf{x}_{1_{cp}}\end{bmatrix} = \mathbf{\Phi}^H\mathbf{\Phi}\mathbf{d} - \mathbf{\Phi}^H\mathbf{\Phi}_p\mathbf{d} = (\mathbf{I} - \mathbf{\Psi})\mathbf{d} = \mathbf{\Gamma}\mathbf{d} \tag{17}$$

where $\mathbf{I} \in \mathbb{Z}^{Q\times Q}$ is the identity matrix and $\mathbf{\Psi} \in \mathbb{C}^{Q\times Q}$ is the ICI matrix, which can be found as:

$$\begin{aligned}\mathbf{\Psi}_{(l,n)} &= \mathbf{\Phi}^H_{(l,m)}\mathbf{\Phi}_{p(m,n)} = \frac{1}{Q}\sum_{m=0}^{Q-1} e^{\left(\frac{-j2\pi lm}{Q}\right)}e^{\left(\frac{j2\pi mn}{Q}\right)}\\ &= \frac{1}{Q}\sum_{m=\gamma Q-Q/4}^{Q-Q/4-1} e^{\left(\frac{-j2\pi lm}{Q}\right)}e^{\left(\frac{j2\pi mn}{Q}\right)}\\ &= \frac{1}{Q}\sum_{v=\Psi Q}^{Q-1} e^{\left(\frac{-j2\pi l(v+Q/4)}{Q}\right)}e^{\left(\frac{j2\pi (v+Q/4)n}{Q}\right)}\\ &= \frac{e^{(-j\pi(l-n))}}{Q}\sum_{v=\gamma Q}^{Q-1} e^{\left(\frac{-j2\pi(l-n)}{Q}\right)}\\ &= \begin{cases}1-\gamma, & l = n\\ \frac{e^{(-j\pi(l-n))}}{Q}\sum_{v=\gamma Q}^{Q-1} e^{\left(\frac{-j2\pi(l-n)}{Q}\right)}, & l \ne n\end{cases}\end{aligned} \tag{18}$$

where, the size of CP part in IEEE 802.11 systems is equal to the quarter of the symbol length, hence $\mu = Q/4$, thus in

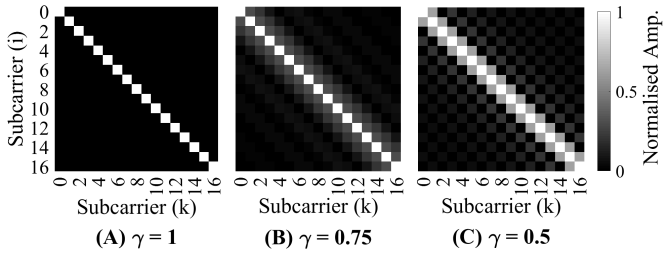


Fig. 6. Heatmap of the covariance matrix that depicts the interference between neighbouring subcarriers. (A) is for orthogonal symbols; (B) and (C) are for partial symbols with $\gamma = 0.75$ and 0.5, respectively. The normalised amplitude indicates the interference coefficient amplitude normalised to the subcarriers represented in the main diagonal.

the above equation it is used $Q/4$. Next, the covariance matrix ($\mathbf{\Gamma} = \mathbf{I} - \mathbf{\Psi}$) is given by:

$$\mathbf{\Gamma}_{(l,n)} = \begin{cases} \gamma, & l = n \\ \frac{-e^{(-j\pi(l-n))}}{Q} \sum_{v=\gamma Q}^{Q-1} e^{\left(\frac{-j2\pi(l-n)}{Q}\right)}, & l \neq n \end{cases} \tag{19}$$

To visualise the effect of interference, resulting from the truncation in time, on a received partial symbol, Fig. 6 (A) shows that all the subcarrierrs are orthogonal with no interference while (B) and (C) in the same figure show that the interference caused on a specific subcarrier (represented on the main diagonal) arises from the neighbouring subcarriers (on the secondary diagonals) when γ is smaller than 1.

6) Partial Symbol Detection With Sphere Decoder: In order to recover the bits of a partial symbol, the covariance matrix, $\mathbf{\Gamma}$, is used in a Sphere decoder, to retrieve the data of a partial symbol. The Sphere decoder mathematical representation is given as:

$$\hat{\mathbf{d}}_{\mathbf{sd}} = \arg \min_{\mathbf{d} \in M^Q, \|\tilde{\mathbf{y}} - \mathbf{\Gamma d}\|^2 \leq g} \| \tilde{\mathbf{y}} - \mathbf{\Gamma d} \|^2 \tag{20}$$

where $\hat{\mathbf{d}}_{\mathbf{sd}} \in \mathbb{C}^{Q \times 1}$ is the estimated data bits vector in the partial symbol, M^N is the number of all possible signal combinations in a symbol, M is the cardinality of the constellation diagram, Q is the number of subcarriers and g is the radius of the hypersphere search area that is centred at $\tilde{\mathbf{y}} \in \mathbb{C}^{Q \times 1}$, which is the demodulated partial symbol after equalisation and phase compensation, $\mathbf{d} \in \mathbb{C}^{Q \times 1}$ is the complex input data vector that has the highest probability of being transmitted. The solution is found by finding the minimum Euclidean distance from all the possible vector combinations which take place inside the multi-dimensional hypersphere search area. This results in data recovery of the partial symbol.

B. Physical Layer Design

PSR uses the 802.11a standard for preamble and data generation in the physical frame, but with a receiver modified with added PSR decoder and a triggering mechanism. The PSR decoder is a nested-loop decoder, consisting of (i) an *inner-loop decoder* and (ii) an *outer-loop decoder*. The inner-loop decoder is designed to recover a single symbol partially corrupted by interference, by applying a specialised decoder optimised to act on the collision-free part of the single symbol (see **(b)** in Fig. 3); this will be explained in Section III-B.1. The outer-loop decoder recovers the rest of a user's symbols following a specially designed interference cancellation technique as in Section III-B.2.

1) Inner-Loop Decoder: The CP part of the received partial symbol is cleaned from the ISI components coming from the previous symbol. This is explained using mathematical representations in section III-A.3 and shown in Fig. 7 part (a). Then, the partial symbol, including the ISI-free CP part, is passed to the next decoding stage to recover its data. The decoding stage is implemented using an iterative receiver that jointly equalises channel effects and phase offset, and cancels interference in every iteration, as shown in Fig. 7 parts (b–f).

a) The flow of an inner-loop decoding iteration: The flow of the decoding process, for a single partial symbol, in the first iteration of the joint iterative decoder, is the following: (i) as there is no information of the estimated multi-path components, the symbol is directly passed to the channel equalisation and phase compensation processes (shown in Fig. 7 part (b)), (ii) as there is no information to estimate the subcarrier interference, the equalised partial symbol is passed to a set of Sphere decoder to recover its data bits, where these bits are de-interleaved and decoded to recover the original transmitted bits before channel encoding, $\hat{b}$, as shown in Fig. 7 part (c), (iii) these bits, $\hat{b}$, are encoded, interleaved and mapped to reconstruct a full symbol, then the latter is used to estimate the subcarrier interference, which is used in the next iteration (steps ii and iii are further detailed in the next paragraph), and (iv) the full symbols is also used to estimate the missing multi-path components, which is used in the next iteration. The processes in step (iv) are represented mathematically in section III-A.4 and depicted in Fig. 7 part (e). After that, for iteration > 1, the differences between the first iteration and the rest of the iterations are: (i) the missing multi-path components are known and hence they are added to the partial symbol before the channel equalisation, as given in equation (13) and shown in Fig. 7 part (f), and (ii) the subcarrier interference is also known and hence it is subtracted from the equalised partial symbol, as explained in the next paragraph and depicted in Fig. 7 part (d), before being passed to the data bits recovery stage, as shown in Fig. 7 part (c). Repetition of these joint processes in each iteration leads to the enhancement of the error rate performance of the decoding process.

b) Partial symbol detection mechanism and discussion: In Section (III-A.6) we provide the details of how to use the covariance matrix with the sphere decoder to recover the data bits from the partial symbol. We note, however, that using a single Sphere decoder, to recover the data in all subcarriers, leads to high computational complexity, hence we adapt the method of [11], in which the subcarriers are divided into sets of blocks, each is operated on by a smaller, hence more computationally efficient, Sphere decoder. For 802.11a signals, there is a natural division into such blocks (of subcarriers) and separation of these (5 blocks) by the four pilot tones, which are placed between the data subcarriers.

The iterative mechanism Due to using a set of Sphere decoders, the interference among subcarriers, in the partial

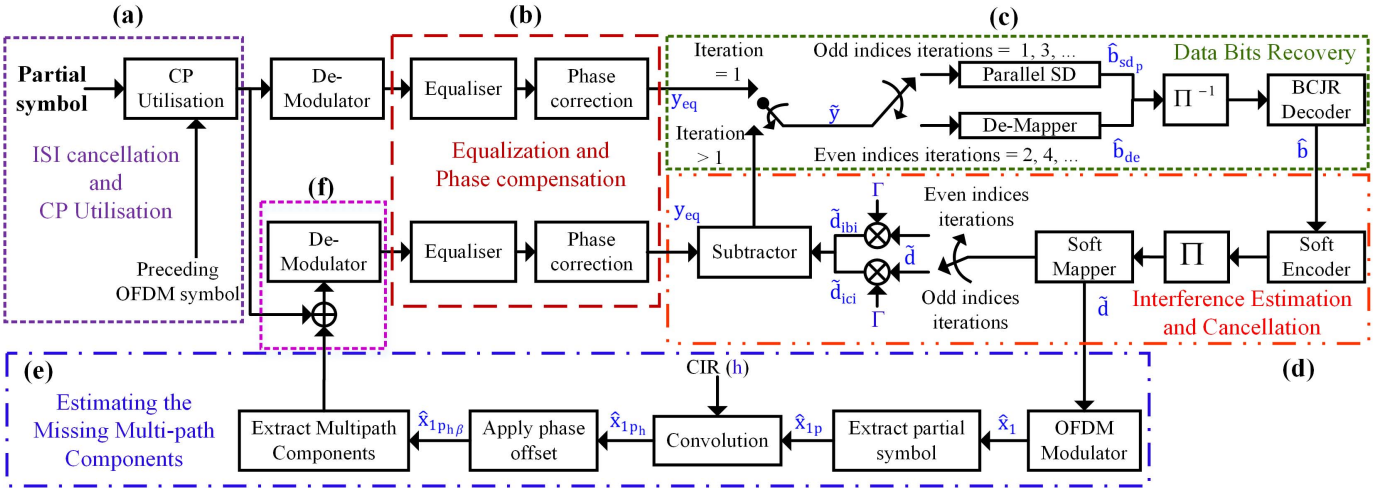


Fig. 7. The proposed Inner-Loop Decoder eliminates ISI and uses information in the CP then equalises and corrects phase; recovers the bits from partial symbols; re-encodes the symbols to estimate and subtract the interference of the original signal before passing the result to the next iteration.

symbol, can be identified and modelled as two types of interference. In our design, the two interference types have to be cancelled iteratively. The two interference types are: (i) interference within a block of subcarriers (intra-block interference or inter-carrier interference), reduced with the first iteration and reduced further with each odd iteration and (ii) interference between subcarriers of adjacent blocks (inter-block interference) which is eliminated iteratively starting with the second iteration and then on every even iteration.

The odd indices iterations In order to recover the data in each block of subcarriers, we calculate and use the covariance matrix of each block (i.e. apply $\mathbf{\Gamma}$ with indicies of that block), in a small-size Sphere decoder set following equation (20). The mathematical derivation of the covariance matrix (Γ) is given in section III-A.5. The output of the small-size Sphere decoder set ($\hat{\mathbf{b}}_{\mathbf{sd_p}}$) is de-interleaved and decoded using soft-output BCJR decoder (see part (c) in Fig. 7). Next, the soft-bits ($\hat{\mathbf{b}}$) are encoded using soft-encoder where its output bits are interleaved and mapped ($\tilde{\mathbf{d}}$). Then, the interference in each block ($\tilde{\mathbf{d}}_{\mathbf{ici}}$) is estimated and then cancelled, as in part (d) in Fig. 7. After that, the interference components are subtracted from the equalised partial symbol ($\mathbf{y_{eq}}$) before the symbol being passed to the next iteration, as in part (d) in Fig. 7.

The even indices iterations After cancelling the interference within the blocks in the odd indices iterations, here in the even indices iterations, the decoder cancels the inter-block interference among the subcarrier blocks. To achieve this, the received signal from the previous iteration is fed to a soft-output demapper, which gives LLR values of the received symbols ($\hat{\mathbf{b}}_{\mathbf{de}}$). Then, the output is de-interleaved and decoded using soft-output BCJR to get soft-bits, $\hat{\mathbf{b}}$, as in part (c) in Fig. 7. After that, the inter-block interference ($\tilde{\mathbf{d}}_{\mathbf{ibi}}$) is estimated and subtracted before the symbol is being passed to the next iteration. Repetition of these processes results in decoding efficiency enhancement.

2) Outer-Loop Decoder: While the inner-loop decoder recovers the data transmitted in a symbol partially corrupted by collisions, the outer-loop decoder uses the information of one symbol to help detect other symbols in a predefined length of symbols, for example in a frame. Fig. 8 illustrates the operation of the outer-loop decoder. Starting with the received (R_x) signal, which comprises two overlapped signals of the two users; each one of these is composed of a set of OFDM symbols, represented by the numbered symbols in the figure, where those with odd indices belong to user 1. For user 1, the ($\mathbf{r}_{1p}$) is the collision-free partial symbol, while the rest of the symbols (of both users) are all corrupted by interference from user 2. After the inner-loop decoder successfully recovers the data bits of $\mathbf{r}_{1p}$, the outer-loop decoder reconstructs symbol 1 ($\mathbf{r}_1$). Then, the outer-loop decoder acts on $\mathbf{r}_1$ and the rest of the symbols, so that all symbols with interference are recovered.

The outer-loop decoding operation is set in different stages and starts with a decoding window of a length equal to one symbol as shown in Fig. 8 in stage $\#\mathbf{I}$. After this initial stage, the decoding window size is increased linearly with the number of stages, until it reaches the size of G symbols, which is the maximum number of symbols that can be detected for a given user in a single outer-loop decoding process. For this work, the value of G was selected empirically following an extensive number of experiments (Section V-C). The outer-loop decoding operates a sliding window algorithm, where in each window data recovery is operated to recover data only of that particular window's first symbol of each user. After that, the two users' two recovered symbols are used to cancel interference by subtracting them from the received overlapped signal and the result is passed to the next decoding window. This is elaborated by considering the set of stages below.

a) Initial stage of decoding; stage $\#\mathbf{I}$: Here, the outer-loop decoder uses the inner-loop decoder (described in Section III-B.1) to decode the partial symbol, $\mathbf{r}_{1p}$, and recover its data bits. After that, the outer-loop decoding uses the recovered data bits of $\mathbf{r}_{1p}$ to reconstruct the full symbol $\mathbf{r}_1$ in the time domain. The symbols reconstruction process includes passing the recovered bits into a convolutional encoder, an interleaver, a mapper and finally an OFDM modulator, giving the reconstructed $\mathbf{r}_1$, which is passed to the next stage.

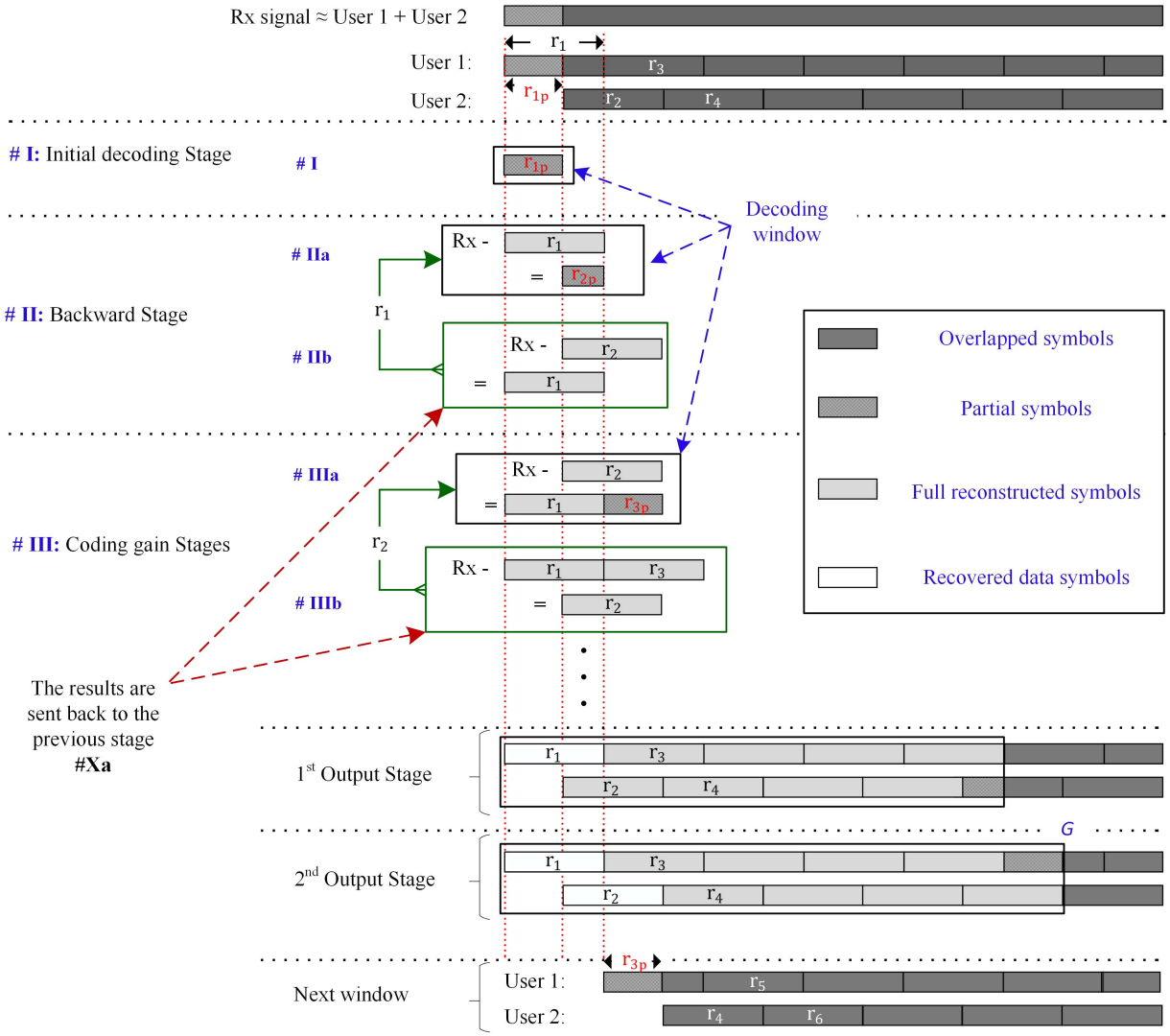


Fig. 8. The outer-loop decoding algorithm recovers the received (R_x) signal, which comprises the overlapped user 1 and 2 signals. The algorithm uses the inner-loop decoder to recover the data in the collision-free portion of symbol 1 ($\mathbf{r}_{1p}$) in the first stage #**I**. In the backward stage, the reconstructed symbol 1 ($\mathbf{r}_1$) is subtracted from the R_x signal to obtain partial symbol 2 ($\mathbf{r}_{2p}$) (stage #**IIa**), which is reconstructed to symbol 2 ($\mathbf{r}_2$). The $\mathbf{r}_2$ is subtracted from the R_x signal to obtain symbol 1 ($\mathbf{r}_1$) as an interference-free symbol (stage #**IIb**), which is sent back to stage #**IIa** and repeat #**IIa** and #**IIb** to enhance the interference cancellation. In the coding gain stages, the partial symbol 3 ($\mathbf{r}_{3p}$) is extracted (stage #**IIIa**), then $\mathbf{r}_1$ and $\mathbf{r}_{3p}$ are decoded together to increase the coding gain. After that, the reconstructed symbols 1 and 3 are subtracted from the R_x to yield $\mathbf{r}_2$ as if it was received with no interference (stage #**IIIb**), which is sent back to stage #**IIIa** and repeat #**IIIa** and #**IIIb** to enhance the interference cancellation. Then, the algorithm increases the decoding window size in every additional stage up to stage G in this example, where decisions on symbol 1 and 2 are made. After that, the decoding algorithm starts a fresh decoding window from symbol 3.

b) Backward stage of decoding; stage #**IIa, b***:* The decoding window size is set to two symbols, these are the reconstructed $\mathbf{r}_1$ and user 2's first partial symbol 2 ($\mathbf{r}_{2p}$). This stage is divided into five steps: (i) basic interference cancellation by subtracting the reconstructed $\mathbf{r}_1$ from the received signal (R_x) to obtain an interference-free $\mathbf{r}_{2p}$, as shown in stage #**IIa** in Fig. 8; (ii) in a manner similar to that of the initial decoding stage, the $\mathbf{r}_{2p}$ is decoded using the inner-loop decoding to recover its data bits, then the outer-loop decoding reconstructs the full symbol 2 ($\mathbf{r}_2$); (iii) next, $\mathbf{r}_2$ is subtracted from the received signal to get $\mathbf{r}_1$ with the effect of user 2 interference removed, as shown in stage #**IIb** in Fig. 8; (iv) then $\mathbf{r}_1$ is fully decoded to recover its data and then reconstructed to yield a received $\mathbf{r}_1$, as if there were no interference from user 2; finally, (v) the outer-loop repeats the first and second steps to reconstruct an enhanced interference-free version of $\mathbf{r}_2$ to be passed, together with $\mathbf{r}_1$ to the next stage.

The reason for stage #**II** is that the inner-loop decoder results in higher error rates relative to those of a conventional OFDM decoder, because partial symbols contain fewer signal samples, which means each symbol has lower energy content than the full symbol. Therefore, stage #**II** cancels the mutual interference effects of the two users to obtain interference-free $\mathbf{r}_1$ and $\mathbf{r}_2$, to yield better error rate performance for both users.

It should be noted that the partial symbol length of the two users is most likely to be unequal but the sum of the period

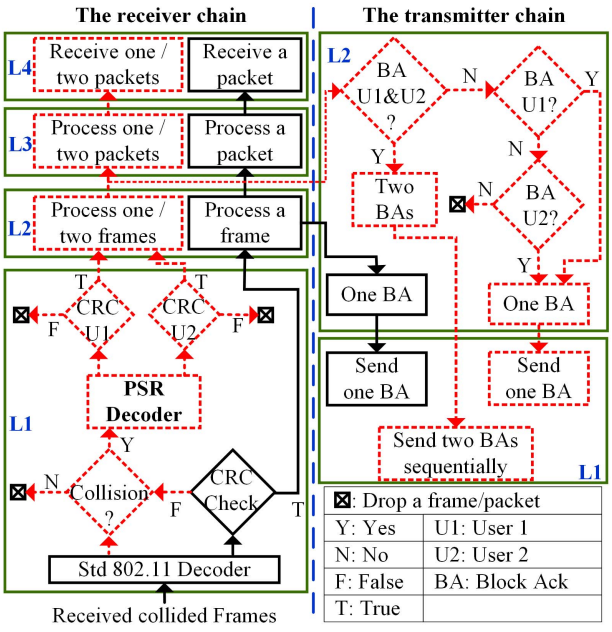


Fig. 9. Schematic chart of Partial Symbol Recovery system. The solid-black lines depict the existing 802.11 receivers, while the dot-red lines represent the additional PSR functions in each layer.

of two colliding partial symbols is equal to the duration of a full symbol.

c) Middle (coding gain) stages of decoding: In the same way as the backward stage, the decoding window increases its size by one symbol for every additional stage until the stage preceding the hard decision, in which, all the symbols in the decoding window are full symbols except for the last symbol. For instance, when the decoding window size is equal to six symbols, there will be five full symbols and one partial symbol. Through these stages, the symbols of each user go through the appropriate convolutional decoder, to improve the coding gain towards the optimum gain of the used convolutional code.

d) Hard decision on symbol 1: The outer-loop decoder gives the first output, which is the recovered data bits of $\mathbf{r}_1$.

e) Final stage of decoding: The outer-loop decoding produces the second output that is the recovered bits of $\mathbf{r}_2$.

f) Reset the decoding window: Recovered $\mathbf{r}_1$ and $\mathbf{r}_2$ are reconstructed and subtracted completely from the overlapped received signal before being passed to the next decoding window, also of size G. The above processes are repeated again but now starting with a fresh decoding window, of size equal to one, to decode partial symbol 3 ($\mathbf{r}_{3p}$) and ending with a decoding of full symbols 3 and 4 ($\mathbf{r}_3$ and $\mathbf{r}_4$, respectively). This carries on to the end of the frame, making hard decision on two symbols only per decoding window.

C. Link Layer Design

In the current 802.11 receivers, the receiving station transmits an acknowledgement (ACK) after a data frame is successfully decoded and passes the cyclic redundancy check (CRC), as shown in Fig. 9. Otherwise, the received frame is dropped and no ACK is sent. The solid-black lines of Fig. 9 depict this process, in the existing 802.11 receivers. On the other hand, our design modifies the data link layer by adding a new acknowledgement scheme, which allows the receiving station to acknowledge successfully recovered frames of the two users, as shown by the added dotted-red blocks in Fig. 9. In this system design, we use delayed block acknowledgement (BA) mechanism from 802.11n to delay sending the BA to the transmitting users until PSR fully decodes the two colliding frames. This is because one of the colliding users could finish transmitting data frames while the other user has not yet completed its transmission.

Decoding overlapped frames necessitates two different designs; (i) for intra-network collision (the scenario in Fig. 2 (A)) and (ii) for collisions in two co-existing networks (the scenarios in Fig. 2 (B), (C1), (C2) and (D)). We note that, at the physical layer, the two designs are identical.

1) Intra-Network Collisions: When two users are transmitting concurrently, the receiving station of the two users decodes the overlapping data frames using PSR as explained in Section III-B. However, the acknowledgement mechanism differs according to whether the collisions occur (i) between data frames of two users or (ii) between data and control frames.

a) Collision between data frames of two users: Fig. 10 (A) depicts the operation of the AP using standard 802.11 and PSR receivers. The 802.11 receiver sends a BA indicating indices of successfully recovered frames. On the other hand, after successful recovery of the overlapping frames using PSR, the AP sends two separate BAs, one for each user. Each BA indicates the indices of the recovered frames of each transmitter, as shown in Fig. 10 (A).

b) Collision of data frames with control frames: In this case, exemplified by Fig. 10 (B), user 2 sends control frames to the AP, which collides and hence no ACK is received. Then user 2 re-sends the control frame after a random wait, in the worst case and under heavy traffic from user 1, further collisions may occur. In the 802.11 receiver, the affected data frames of user 1 are dropped and a single BA is sent for the recovered frames. In contrast, for the PSR case, overlapped frames are recovered for both users and the receiver waits until no transmission is detected, then an Ack is sent for the control of user 2 and a BA for the data frames of user 1.

2) Inter-Network Collisions: In this case of inter-network scenarios, the overlapped frames at the recipient station have different destination addresses. Therefore, after decoding the overlapped frames, the receiving station only takes the frames that have its address in the destination field. Hence, sending only a single BA after a complete reception. Fig. 10 (C) depicts the process of sending a single BA. The advantage of using PSR over standard 802.11 receivers, is the ability to recover overlapped frames. This enhances the throughput and reduces the retransmission rate.

D. System Parameters Identification and Estimation

1) Collision Detection and PSR Activation: We follow a technique similar to Zigzag Decoding [1], where the known

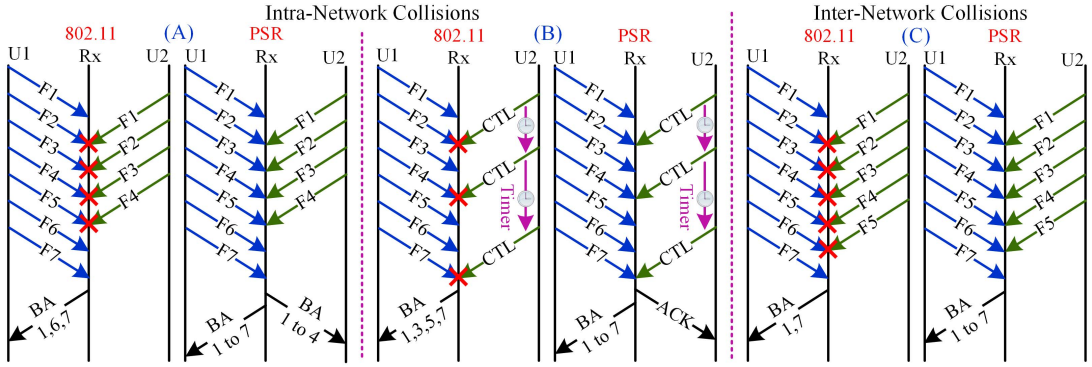


Fig. 10. Message sequence chart of intra- and inter-network collision scenarios. User 1 and user 2 cause collisions at the receiving station when they have concurrent transmissions. (A) collisions of data frames of two users at the AP; (B) collisions between data and control frames at the AP; (C) collisions are the inter-network collision scenario. The character F refers to the data frames, while the CTL refers to the control frames.

preamble is correlated with the received signal. When the two preambles are properly aligned, a correlation peak results. A single incoming frame results in a single correlation peak at the start of that frame. The generation of a second peak within the duration of the first frame indicates that a colliding/interfering frame is present, since the second peak is a result of correlation with the colliding frame's preamble. This identifies the cause of CRC failure as a collision and hence the receiving station activates PSR to decode the colliding frames.

2) γ Estimation: It is important to find the non-interfered part of the first symbol, expressed in the value of γ as defined in Section III-B.1. the value of γ determines the inter-carrier interference levels and hence the covariance matrix, required for the detection process in the inner-loop as explained in Section III-B.1. γ can simply be estimated (see Fig. 3) by finding the time delay between the starting sample, of the colliding user transmission using the first correlation peak and the second correlation peak.

3) User ID, CFO, and CIR Estimation and Compensation: The received signals are distorted by the wireless multi-path channels and the CFO effects and contaminated with AWGN noise. In order to mitigate the multi-path channel and CFO effects, the receiving station estimates the CIR and CFO for each user. In 802.11 standards, such are estimated using the received preamble attached to each frame. This has to be modified for a system assuming collision. The steps a to c below detail the modified operation, which requires refining the estimates to deal with interference and collision.

a) Colliding user ID and initial CFO and CIR estimates: For user 1, the CIR and CFO are estimated using its preamble signal, which is received within the collision-free region. Unfortunately, this can not be done for user 2, since its received preamble is corrupted by the collision with the signal of user 1. To resolve this problem, we introduce a new technique to define the colliding user ID, and the corresponding CIR and CFO. The technique, implemented in two stages, is based on the premise that the combination of CIR and CFO for each station will be unique. The first stage, the receiving station keeps a record of user IDs and associated most recent CIR and CFO of users stations in its vicinity, whenever user data is received with no collision. The second stage is to define which set of CIR and CFO is the correct match to the colliding user. To achieve this, our decoder sequentially convolves all the saved CIR with the known preamble signal and then applies CFO. After that, the decoder correlates the resulting convolution values with the received signal and compares the results of all correlation processes. This comparison identifies the colliding station ID through its most recent CFO and CIR, which are used as an initial estimate to decode the signal of the colliding user.

b) Updating CFO and CIR information: After identifying the initial CFO and CIR information as mentioned earlier, the decoder uses this information to decode the colliding signal until the preamble of the user 2 (the colliding signal) is fully recovered. Then, the extracted/recovered preamble signal of user 2 is used to get a new estimate of the CFO and CIR. In general, any residual error in estimating the CFO will result in a phase offset, which is estimated using the pilot tones inserted in 802.11 OFDM symbols.

c) CFO compensation: We correct CFO for a partial symbol before it is passed to the inner-loop decoder by applying a frequency shift to the signal in the time domain:

$$\mathbf{x}_{1_p} = \mathbf{x}_{1_{p_{cfo}}} \cdot e^{-j2\pi\delta f n} \tag{21}$$

where $\mathbf{x}_{1_p}$ and $\mathbf{x}_{1_{p_{cfo}}}$ are the partial symbol (with sub index 1) after and before compensating CFO, respectively, and the exponential term represents the applied correction of the frequency shift, δf, which is estimated using the preamble, and n the time index from the beginning of the frame.

IV. Implementation

The 802.11 devices for both transmitters and receivers are WARP V3 devices software defined radios (SDRs). The WARP devices operate on channel 17 at 5660 MHz and use TP-link antennas with 5 dBi gain. We set the sampling rate in the WARP device to be 20 MHz. The WARP devices connect to a Linksys SE4008 WRT 8-port gigabit Ethernet switch through Ethernet cables. The network topology is shown in Fig. 11.

A. Physical Layer

The SDR testbed is driven by MATLAB R2017b for 802.11a signal generation, following the 802.11a OFDM physical frame structure. A 64-point IFFT generates the symbols,

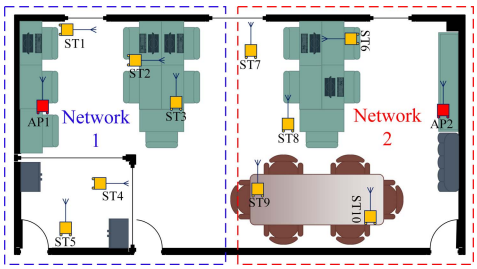


Fig. 11. Indoor experimental testbed map. The testbed consists of ten nodes that operate as user stations (STs) and two nodes that operate as access points (APs). The nodes are distributed across two rooms for two networks in NLOS and LOS indoor environments.

and uses 48 inputs for data, 4 for pilots and the rest are zeros. A preamble is attached at the beginning of each frame.

The received overlapped signal is saved to disk to be processed in off-line decoding. The PSR and SIC decoders are written in Matlab. The decoders save the decoded bits to disk for performance evaluation.

B. Link Layer to Transport Layer

We build the network topology scenario shown in Fig. 11 in NS-3 [29] with the 802.11n standard system specifications. First, we evaluate standard 802.11, in which an ACK is sent to the transmitter only when a frame is decoded correctly. On the other hand, SIC and PSR are built on top of 802.11n, and so act exactly as 802.11n unless there is a collision, in which case, the frame that could not be decoded by the 802.11n receiver is examined to detect if it experiences collision event as explained in Section III-D.1. When a collision event is detected, trace driven data of the decoder under evaluation (either SIC or PSR), which contains details of the decoding performance, is used by the receiver to make a decision on the success or otherwise of frame decoding. The use of NS-3 in our evaluations, is to achieve almost real life collision environment where transmissions are not synchronised, the frame length is variable and the γ variable is not set.

V. Evaluation

In this section, we evaluate the PSR system versus the 802.11 and SIC systems. In Section V-A we describe the methodology and in Section V-B we evaluate the PSR versus SIC in terms of rate region. Experimental evaluation of the PSR system in the link layer is given in Section V-C and of the end-to-end throughput performance is given in Section V-D.

A. Methodology

We evaluate PSR with 12-nodes using our SDRs testbed. Physically, each node is a WARP device connected to a computer. The network topology, shown in Fig. 11, contains both non-line-of-sight (NLOS) and line-of-sight (LOS) links. In order to evaluate PSR with real traffic, which includes realistic collision environment (*i.e.* collisions occurrence are not manually set), in a network that covers up to the transport layer, the network testbed is implemented in two steps: (i) the physical frames collisions intentionally occur in a realistic RF environment experimented in a two-room office, as shown in Fig. 11, to capture the effect of various real channels in PSR decoding capability between two colliding frames, (ii) the data traffic is implemented following a realistic network topology of the 12-node testbed using the experimental collision decoding success for end-to-end evaluation.

1) Experimental Collision Traces: In our application, we focus on decoding two 802.11 frames that have collided. Two nodes simultaneously transmit data to another node to intentionally cause a collision. For example, ST 1 and ST 2, shown in Fig. 11, both transmit data to AP 1, and the received collision data is saved to disk to be decoded.

For each experimental run, we vary the experiment parameters, which are the modulation formats and γ variable. We experiment for BPSK and QPSK modulation formats and for ten values of the fractional overlap, $\gamma = 0, 0.1, \ldots, 0.9$. In each experiment we send 600 physical frames from two nodes and collect 600 collision traces. Every frame carries 12,000 bits (1,500 bytes) and the measured SNR at the receivers is within the range of 13-25 dB. The wide range in SNR arises from the variation in the channel environment, as the two-room office was occupied by eight people. Also, two of the stations, ST 4 and ST 5 as shown in Fig. 11, were positioned in a storage area partitioned with glass walls.

2) Schemes Compared for End-to-End Evaluation: Using our 12-node testbed and the trace driven simulations, we evaluate three schemes end-to-end: (i) PSR, (ii) SIC, and (iii) 802.11n. We evaluate with the following metrics:

- **Rate region comparison**, which compares the rate region of the two user uplink in AWGN channel using SIC and PSR receivers.
- **Frame error rate (FER)**, which is the percentage of incorrectly decoded colliding frames to the total collided frames. In this evaluation, a frame is considered successfully decoded if it passes the CRC.
- **Throughput**, this is the data throughput-per-second measured at the transport layer.
- **Frame retransmission rate**, this is the number of physical frame retransmissions-per-second measured at L2.

B. Rate Region Comparison

In this section, we compare the rate region of the two users uplink AWGN channel, using SIC and PSR. Fig. 12 **(A)** shows the rate region of both SIC and PSR systems using a data rate range of 3-6 Mbps, where the first modulation and coding scheme 0 (MCS0) of 802.11a systems is used in this evaluation. The data rate of users is varied by changing the signal power of a user, while the signal power of the other user and the noise power level is kept constant. The SNR for this test is equal to 2.3 dB, which is the lowest SNR that achieves the optimum rate region using SIC receivers.

The current maximum rate region of a communication system, for a two-user scenario, can be achieved using SIC receivers [3]. In 802.11a systems, this is depicted by the black curve in Fig. 12 **(A)**. The maximum rate of a user in this rate region is constrained by the rate of the other user who shares

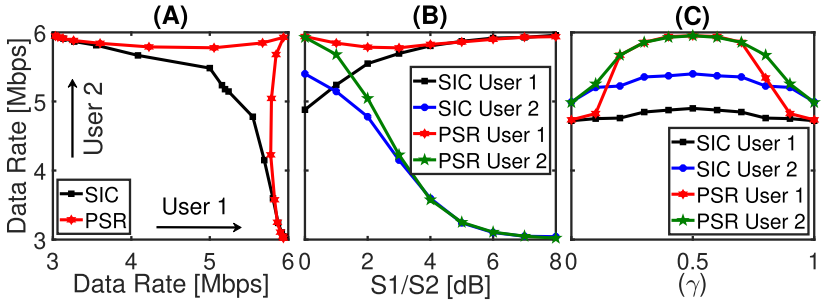


Fig. 12. Rate region comparisons. (A) the rate region of the two user uplink AWGN channel. (B) The data rate of user 1 and 2 using SIC and PSR. (C) Rate performance versus different values of γ.

the same link. This is because a SIC decoder considers the interfering signal of a user as noise in order to decode the signal of the other user [3]. Interestingly, this role does not apply to the PSR system. This is because the PSR exploits the collision-free partial symbol to bootstrap the colliding frames in a collision, where such a partial symbol is not constrained by the rate of the interfering user. Therefore, the rate region achieved by the PSR receiver for two users is beyond that of the SIC receivers, as shown by the red curve in Fig. 12 **(A)**.

To investigate further how the PSR decoder can outperform the rate region achieved by the SIC decoder, we provide the rate performance of the two users, using SIC and PSR with $\gamma = 0.5$, versus the difference between their signal powers. This is depicted in Fig. 12 **(B)**. As expected, using SIC decoder, the rate of a user decreases as the power difference between the two received signals decreases. On the other hand, the PSR decoder outperforms the SIC decoder, where the rate performance of a user is maintained at almost the same level regardless of the difference in received signals power.

However, the performance of the PSR is not without limitation, where it depends on the duration of the partial symbols in time; such duration is represented by the γ ratio factor, defined in Section III-B.1. Fig. 12 **(C)**, shows the rate performance of the two users using the PSR and the SIC decoders versus different values of γ measured at $S_1/S_2 = 0$ dB. The figure shows that the PSR performance outperforms the SIC decoder for most of the γ values and then converges to the SIC performance level at the edges of γ values.

C. FER Measurements in Link Layer

The FER performance evaluation for SIC and PSR is carried out using all permutations of two nodes transmitting simultaneously to another node in our testbed (this is called collision cases). In total our indoor testbed, shown in Fig. 11, provides 45 collision cases. The evaluation metrics are (i) FER versus the decoding window size of the outer-loop decoder, where the decoder is described in Section III-B.2; (ii) FER versus different values of γ; and (iii) the CDF of the FER for PSR and compared to SIC for all the collision cases.

a) Performance evaluation of the outer-loop decoder: As mentioned in Section III-B.2, the PSR makes a decoding decision when its outer-loop decoder reaches the maximum decoding window. In this experiment, the decoding window size is set to $G = 10$, G defined in Section III-B.2. Fig. 13

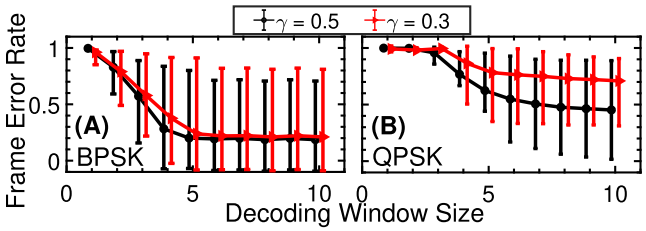


Fig. 13. Average FER of all collision cases versus decoding window size of the outer-loop decoder. Maximum and minimum FER values, due to variation of channel conditions and noise, indicated by the vertical lines. (A) using BPSK modulation; (B) using QPSK modulation.

shows the FER performance of different stages of the outer-loop decoder. Each marker point in the curve represents the average FER performance of one of the 45 collision cases tested, while the candle bars indicate the highest and lowest FER performance for each case. Such a range of variation for each case is due to differing channel effects, CFO and signal to noise level.

Also from Fig. 13, it is clear that PSR with BPSK modulation is better than that of using QPSK modulation. This is because having a partial symbol results in ICI exclusively from the real part of the signal to the imaginary part and vice versa, as well as, inter-carrier interference from the real and imaginary parts to themselves. Thus, a one-dimensional modulation format, such as BPSK, will be decoded with lower error rate, using the inner-loop decoder compared to a two-dimensional modulation, such as QPSK.

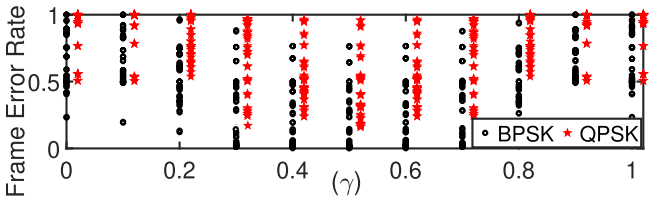
b) FER versus γ: Fig. 14(a) shows FER performance of PSR for different collision cases versus γ values, which is taken from the last stage of the outer-loop decoder, $G = 10$. Each point in the figure represents the FER of a single collision case (black points are BPSK while red are QPSK). Clearly, the FER performance is almost symmetrical around $\gamma = 0.5$, because the FER performance is restricted to the shortest partial symbol duration of the two colliding users (see Section III-B.2).

c) CDF of FER: The CDF of the FER performance of each collision case is shown in Fig. 14(b), where each marker point on the curves indicates average PSR performance over all γ values. The horizontal lines attached to each point represent the difference in performance of the PSR and SIC decoders.

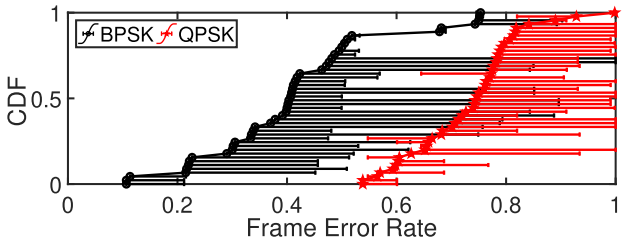
TABLE I

PERFORMANCE SUMMARY OF PSR, SIC AND 802.11A DECODING CAPABILITY OF COLLIDING FRAMES

Experimental set	802.11a		SIC		PSR	
	BPSK	QPSK	BPSK	QPSK	BPSK	QPSK
Decoding capability of colliding frames	0%	0%	37%	9%	59%	25%



(a) Frame error rates vs γ



(b) CDF of frame error rates across links

Fig. 14. PSR FER performance. (a) FER versus γ for different collision cases; (b) CDF of FER across all collision cases (in total 45 cases), each case is a single marker point. The horizontal lines show the difference in FER performance between PSR and SIC.

On average, PSR achieves 22% and 16% better (lower) FER than SIC, using BPSK and QPSK modulation, respectively.

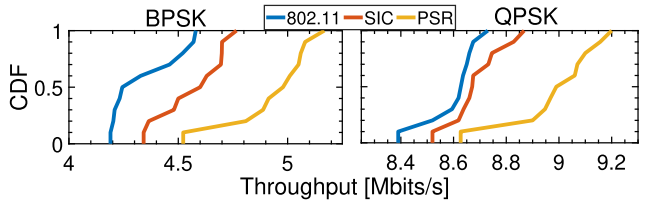
To sum up, Table I shows the decoding capability of colliding frames, which the ability of decoding frames with 12,000 bits, using 802.11a, SIC and PSR receivers. From these results, the PSR receiver is experimentally shown to be effective for non-line-of-sight and line-of-sight indoor environment, thus paving the way for practical implementation of PSR in wireless LAN systems.

D. End-to-End Performance

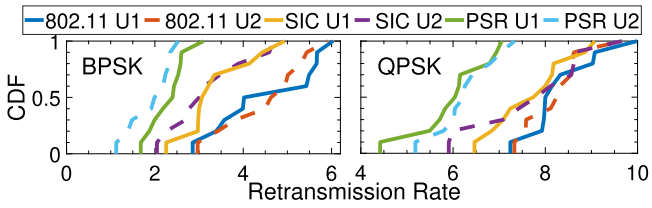
In this section, we use the testbed in Fig. 11 to evaluate the throughput and retransmission rate performance of PSR in intra- and inter-network collisions, described in Section III-C. The two scenarios are implemented in NS-3, driven by our experimental testbed decoding performance.

1) Intra-Network Collisions: We use this case study scenario to evaluate the performance of PSR, SIC and 802.11n for intra-network collisions, such as the collisions from hidden terminal nodes. We use Network 1 in Fig. 11 as the experimental testbed, providing ten collision cases. Data flow of UDP packets using 802.11n frame structure is simulated in NS-3. Request-to-send/clear-to-send (RTS/CTS) is disabled. The network performance is evaluated using throughput and retransmission rate.

a) Throughput performance at the transport layer: We start by comparing the throughput of each decoder. Fig. 15(a) shows the CDF of throughput measured for collision cases in

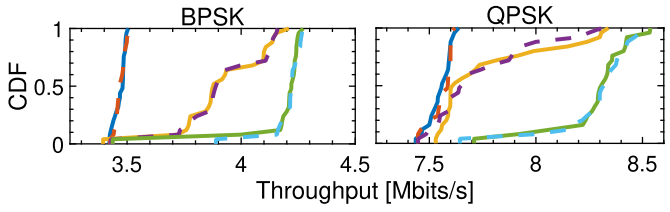


(a) CDF of throughput

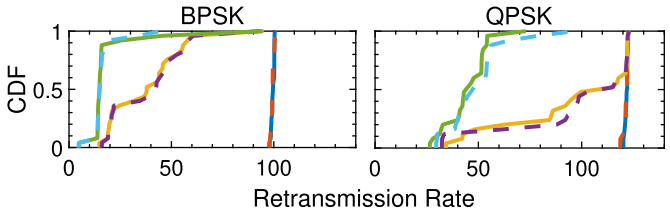


(b) CDF of retransmission rate

Fig. 15. CDF of throughput and retransmission rate in intra-network scenario using uplink-UDP tranmissions. The legends in figure (b) apply to the following figures.



(a) CDF of throughput



(b) CDF of retransmission rate

Fig. 16. CDF of throughput and retransmission rate in inter-network scenario using bulk TCP transmissions.

Network 1. Compared to a baseline 802.11n decoder, PSR improves average throughput by 13.6%, while the average throughput improvement of the SIC decoder is 5.2%. For QPSK modulation, the average throughput improvements of PSR and SIC are 4.7% and 1.2% over the 802.11n baseline. The throughput enhancement comes from the fact that PSR attempts to decode collided frames and recovers their data.

b) Retransmission rate performance: Fig. 15(b) shows the CDF of retransmission rates measured for each collision case in Network 1. PSR and SIC reduce average BPSK retransmission rate by 49.3% and 25.5% respectively. For

TABLE II
SUMMARY OF PSR PERFORMANCE ENHANCEMENT IN COMPARISON TO 802.11N

Experimental set	Uplink UDP		Bulk TCP		Video-on-demand	
	BPSK	QPSK	BPSK	QPSK	BPSK	QPSK
Throughput enhancement	13.6%	4.7%	21.27%	9.7%	21.26%	7.7%
Retransmission rate reduction	49.3%	27.6%	81%	63%	80.8%	57%

QPSK modulation, those reduction change to 27.6% and 7.2%, compared to 802.11n.

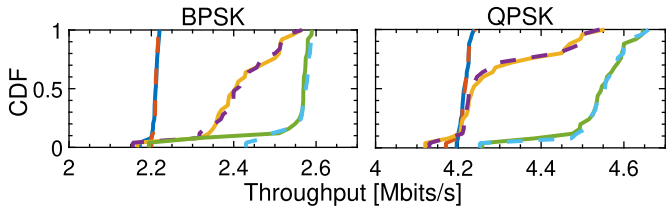
In terms of power consumption, the work in [30] shows that the impact of retransmission on power consumption is directly proportional. Hence, implementing PSR in the current 802.11 systems will decrease the power consumption, occurs in the retransmission scenarios, in the user station by the factor of the retransmission reduction rate, which are 49.3% and 27.6% for BPSK and QPSK modulation formats, respectively, compared to the baseline 802.11n. Furthermore, these results will hopefully inspire the future exploration of the adoption of PSR in the uplink regime of the 802.11ah IoT networks [31]. Thus, the power consumption in the IoT devices can be reduced, due to the reduction in the frames retransmission, and hence, maximising battery and saving costs.

2) Inter-Network Collisions: We evaluate the PSR performance in a scenario representing a topology, where two stations downloading data using (i) bulk TCP and (ii) video-on-demand data flows from the AP in their respective networks, as shown in Fig. 2. Since the two data flows require uplink and downlink data, collisions occur when a station is receiving a frame while the other station, in the neighbouring network, is transmitting a frame to its AP.

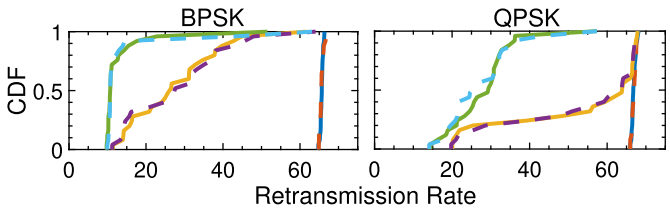
Under these circumstances, we evaluate the downlink performance of the two networks in Fig. 11 as the experimental testbed, providing 25 collision cases for each network. In every experimental run, a station from each network is involved, as well as its AP, hence, there are two links under test: the first link is between a Network 1 station and AP 1 while the second link is between a Network 2 station and AP 2. We measure the throughput at the transport layer of each station, as well as the link layer retransmission rate of AP 1 and AP 2. We plot the CDF of throughput and retransmission rates for both bulk TCP and video-on-demand using BPSK and QPSK modulation formats in Fig. 16 and 17.

a) Throughput performance: According to our experimental results, the average throughput improvements in comparison to 802.11n, shown in Fig. 16(a) and 17(a), are 21.27% and 21.26% for the downlink bulk TCP and downlink video-on-demand, respectively, for BPSK modulation. The average throughput enhancement using QPSK modulation are 9.7% and 7.7% for the above respective topology scenarios shown in the same figures.

The throughput enhancement, in PSR systems, is due to the fact that PSR acts only on the collided frames with decoding capability of 59% and 25% of the collided frames for BPSK and QPSK, respectively, as mentioned in Section V-C. Thus, PSR always has potential for the throughput improvement but never degrades the system performance. This makes PSR an attractive potential system in highly dense user environment, such as massive offices that has multiple 802.11 networks and suffer from inter-network collisions.



(a) CDF of throughput



(b) CDF of retransmission rate

Fig. 17. CDF of throughput and retransmission rate in inter-network scenario using video-on-demand transmissions.

b) Retransmission rate performance: The CDF of retransmission rate comparison is shown in Fig. 16(b) and 17(b). The average retransmission rate in BPSK modulation is reduced by 81% and 80.8% for the downlink bulk TCP and downlink video-on-demand, respectively. The reductions for QPSK modulation are 63% and 57% for the above respective topologies. To summarise the results of PSR system, Table II shows a summary of the performance enhancement in terms of average throughput and retransmission rate compared to the baseline 802.11n systems.

VI. CONCLUSION

This work presents a new receiving technique and design, termed Partial Symbol Recovery (PSR), that allows efficient operation of fully overlapping data frames in wireless communication systems with collision environments. The PSR technique is based on recovery of the parts of OFDM symbols, which are collision-free, followed by the reconstruction of complete symbols to recover progressively the frames of two users suffering collision. The system is evaluated in a testbed of 12-nodes using software defined radio platforms. Extensive experimental results show 10–21% throughput enhancement in 802.11n systems when downlink bulk TCP, downlink video-on-demand and uplink UDP are operated in a collision-rich environment and under different scenarios.

REFERENCES

[1] S. Gollakota and D. Katabi, "ZigZag decoding: Combating hidden terminals in wireless networks," in *Proc. ACM SIGCOMM Conf.*, Aug. 2008, pp. 159–170.

[2] K. Jamieson and H. Balakrishnan, "PPR: Partial packet recovery for wireless networks," in *Proc. SIGCOMM*, 2007.
[3] D. Tse and P. Viswanath, *Fundamentals of Wireless Commu*. Cambridge, U.K.: Cambridge Univ. Press, 2005.
[4] R. Ahlswede, "Multi-way communication channels," in *Proc. 2nd Int. Symp. Inf. Theory*, Thakadsor, Armenian, Sep. 1971, pp. 103–135.
[5] H. Liao, "A coding theorem for multiple access communications," in *Proc. IEEE Int. Symp. Inf. Theory*, Asilomar, CA, USA, Feb. 1972.
[6] M. O. Damen, H. E. Gamal, and G. Caire, "On maximum likelihood detection and the search for the closest lattice point," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2389–2402, Oct. 2003.
[7] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. Inf. Theory*, vol. 48, no. 8, pp. 2201–2214, Aug. 2002.
[8] D. Barry, E. Lee, and D. Messerschmitt, *Digital Communication*, 3rd ed. New York, NY, USA: Springer, 2003.
[9] J. Mazo, "Faster-than-Nyquist signalling," *Bell Syst. Tech. J.*, vol. 54, pp. 1451–1462, Oct. 1975.
[10] I. Kanaras, A. Chorti, M. Rodrigues, and I. Darwazeh, "Spectrally efficient FDM signals: Bandwidth gain at the expense of receiver complexity," in *Proc. ICC*, 2009, pp. 1–6.
[11] T. Xu and I. Darwazeh, "Multi-band reduced complexity spectrally efficient FDM systems," in *Proc. CSNDSP*, Jul. 2014, pp. 982–987.
[12] F. Rusek and J. Anderson, "The two dimensional Mazo limit," in *Proc. ISIT*, 2005, pp. 970–974.
[13] W. Ozan, K. Jamieson, and I. Darwazeh, "Truncating and oversampling OFDM signals in white Gaussian noise channels," in *Proc. CSNDSP*, 2016, pp. 1–6.
[14] E. Hamed, H. Rahul, and B. Partov, "Chorus: Truly distributed distributed-MIMO," in *Proc. SIGCOMM*, 2018, pp. 461–475.
[15] C. Shepard *et al.*, "Argos: Practical many-antenna base stations," in *Proc. MobiCom*, 2012, pp. 53–64.
[16] O. Avner and S. Mannor, "Multi-user communication networks: A coordinated multi-armed bandit approach," *IEEE/ACM Trans. Netw.*, vol. 27, no. 6, pp. 2192–2207, Dec. 2019.
[17] S. Jafar *et al.*, "Degrees of freedom of the MIMO X channel," *IEEE Trans. Inf. Theory*, vol. 54, no. 1, pp. 151–170, Jan. 2008.
[18] S. Gollakota *et al.*, "Interference alignment and cancellation," in *Proc. SIGCOMM*, 2009, pp. 159–170.
[19] C. Suh and D. Tse, "Interference alignment for cellular networks," in *Proc. Allerton*, 2008, pp. 1037–1044.
[20] T. Xu, C. Masouros, and I. Darwazeh, "Design and prototyping of hybrid analog–digital multiuser MIMO beamforming for nonorthogonal signals," *IEEE Internet Things J.*, vol. 7, no. 3, pp. 1872–1883, Mar. 2020.
[21] X. Guo, Y. He, X. Zheng, L. Yu, and O. Gnawali, "ZigFi: Harnessing channel state information for cross-technology communication," *IEEE/ACM Trans. Netw.*, vol. 28, no. 1, pp. 301–311, Feb. 2020.
[22] S. Wong, H. Yang, S. Lu, and V. Bharghavan, "Robust rate adaptation for 802.11 wireless networks," in *Proc. MobiCom*, 2006, pp. 146–157.
[23] M. Vutukuru, H. Balakrishnan, and K. Jamieson, "Cross-layer wireless bit rate adaptation," in *Proc. SIGCOMM*, 2009, pp. 3–14.
[24] J. Perry, P. Iannucci, K. Fleming, H. Balakrishnan, and D. Shah, "Spinal codes," in *Proc. SIGCOMM*, 2012.
[25] K. C. Lin, N. Kushman, and D. Katabi, "ZipTx: Exploiting the gap between bit errors and packet loss," in *Proc. MobiCom*, 2008, pp. 351–362.
[26] J. Choi, M. Jain, K. Srinivasan, P. Levis, and S. Katti, "Achieving single channel, full duplex wireless communication," in *Proc. MobiCom*, 2010, pp. 1–12.
[27] D. Bharadia and S. Katti, "Full duplex MIMO radios," in *Proc. NSDI*, 2014, pp. 359–372.
[28] R. M. Gray *et al.*, "Toeplitz and circulant matrices: A review," *Found. Trends Commun. Inf. Theory*, vol. 2, no. 3, pp. 155–239, 2006.
[29] G. F. Riley and T. R. Henderson, *The ns-3 Network Simulator*, J. Gross, Ed. Berlin, Germany: Springer, 2010, pp. 15–34.
[30] P. Serrano, A. Garcia-Saavedra, G. Bianchi, A. Banchs, and A. Azcorra, "Per-frame energy consumption in 802.11 devices and its implication on modeling and design," *IEEE/ACM Trans. Netw.*, vol. 23, no. 4, pp. 1243–1256, Aug. 2014.
[31] M. Park, "IEEE 802.11ah: Sub-1-GHz license-exempt operation for the Internet of Things," *IEEE Commun. Mag.*, vol. 53, no. 9, pp. 145–151, Sep. 2015.

Waseem Ozan (Member, IEEE) received the B.Sc. degree in electronic engineering from the University of Mosul, Iraq, in 2012, and the M.Sc. and Ph.D. degrees from University College London (UCL), London, U.K., in 2015 and 2020, respectively. He is currently a Senior Research Engineer with MediaTek Wireless Ltd., with focus on radio resource managements standardization for 5G cellular systems. His research interests include reduced capability (RedCap) in 5G systems and ultra-reliable low latency communications (URLLC) in 5G systems.

Izzat Darwazeh (Senior Member, IEEE) received the Graduate degree in electrical engineering from the University of Jordan, Amman, Jordan, in 1984, and the M.Sc. and Ph.D. degrees from The University of Manchester, Manchester, U.K., in 1986 and 1991, respectively.

He currently holds the University of London Chair of communications engineering and leads the 70-Strong Information and Communications Engineering Group, Department of Electronic and Electrical Engineering, University College London, London, U.K., where he is also the Director of the Institute of Communications and Connected Systems. He has authored/coauthored more than 250 articles and book chapters in the areas of optical and wireless communications and monolithic microwave integrated circuits and high-speed/frequency circuits. He coedited *Analogue Optical Fibre Communications* (IEE, 1995) and *Newness Book on Electrical Engineering* (Elsevier) in 2008. He has also coauthored two books *Linear Circuit Analysis and Modeling* (Elsevier, 2005) and *Microwave Active Circuit Analysis and Design* (Academic Press, 2015). In 2002, he proposed (with Miguel Rodrigues) the Fast OFDM concept and in 2003 the SEFDM concept and has been working on these topics since. He also teaches mobile and wireless communications and circuit design. His current research interests include ultra high-speed microwave circuits and wireless and optical communication systems. He is a Fellow of the IET and the Institute of Telecommunications Professionals. He is a Chartered Engineer.

Kyle Jamieson (Senior Member, IEEE) received the B.S. degree in mathematics, computer science, the M.Eng. degree in computer science and engineering, and the Ph.D. degree in computer science from the Massachusetts Institute of Technology in 2008. He is currently a Professor of computer science and an Associate Faculty in electrical and computer engineering at Princeton University. His research interests include mobile and wireless systems for sensing, localization, and communication, and on massively-parallel classical, quantum, and quantum-inspired computational structures for NextG wireless communications systems. He is a Senior Member of the ACM. He then received a Starting Investigator Fellowship from the European Research Council, a Google Faculty Research Award, and the ACM SIGMOBILE Early Career Award. He served as an Associate Editor for the IEEE TRANSACTIONS ON NETWORKING from 2018 to 2020.