ARTICLE IN PRESS

Physical Communication ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at SciVerse ScienceDirect

Physical Communication

journal homepage: www.elsevier.com/locate/phycom

Full length article

Implementation of physical-layer network coding☆,☆☆

Lu Lu [a,b,*], Taotao Wang [a], Soung Chang Liew [a], Shengli Zhang [c]

[a] Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong

[b] Institute of Network Coding, The Chinese University of Hong Kong, Hong Kong

[c] Department of Communication Engineering, Shenzhen University, China

ARTICLE INFO

Article history:
Received 18 April 2011
Accepted 22 February 2012
Available online xxxx

Keywords:
Physical-layer network coding
Network coding implementation
Software radio

ABSTRACT

This paper presents the first implementation of a two-way relay network based on the principle of physical-layer network coding (PNC). To date, only a simplified version of PNC, called analog network coding (ANC), has been successfully implemented. The advantage of ANC is that it is simple to implement; the disadvantage, on the other hand, is that the relay amplifies the noise along with the signal before forwarding the signal. PNC systems in which the relay performs XOR or other denoising PNC mappings of the received signal have the potential for significantly better performance. However, the implementation of such PNC systems poses many challenges. For example, the relay in a PNC system must be able to deal with symbol and carrier-phase asynchronies of the simultaneous signals received from multiple nodes, and the relay must perform channel estimation before detecting the signals. We investigate a PNC implementation in the frequency domain, referred to as FPNC, to tackle these challenges. FPNC is based on OFDM. In FPNC, XOR mapping is performed on the OFDM samples in each subcarrier rather than on the samples in the time domain. We implement FPNC on the universal soft radio peripheral (USRP) platform. Our implementation requires only moderate modifications of the packet preamble design of 802.11a/g OFDM PHY. With the help of the cyclic prefix (CP) in OFDM, symbol asynchrony and the multi-path fading effects can be dealt with simultaneously in a similar fashion. Our experimental results show that symbol-synchronous and symbol-asynchronous FPNC have essentially the same BER performance, for both channel-coded and non-channel-coded FPNC systems.

1. Introduction

In this paper, we present the first implementation of physical-layer network coding (PNC) on the software radio platform. We believe this prototyping effort moves the concept of PNC a step toward reality. Our implementation work also exposes and raises some interesting issues for further research.

PNC, first proposed in [1], is a subfield of network coding [2] that is attracting much attention recently. The simplest system in which PNC can be applied is the two-way relay channel (TWRC), in which two end nodes *A* and *B* exchange information with the help of a relay node *R* in the middle, as illustrated in Fig. 1. Compared with the conventional relay system, PNC could double the

☆ This work is supported by AoE grant E-02/08 and the General Research Funds Project No. 414911, established under the University Grant Committee of the Hong Kong Special Administrative Region, China, and the CUHK Direct Grant 2050464. This work is also supported by NSF of China (Project No. 60902016), NSF of Guangdong (Project No. 10151806001000003), NSF of Shenzhen (Project No. JC201005250034A), and the China 973 Program (Project No. 2012CB315904).

☆☆ This paper was presented in part at the IEEE International Conference on Communication (ICC) 2012, Ottawa, Canada.

* Correspondence to: Department of Information Engineering, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong. Tel.: +852 3943 8382; fax: +852 2603 5032.

E-mail addresses: ll007@ie.cuhk.edu.hk (L. Lu), wtt011@ie.cuhk.edu.hk (T. Wang), soung@ie.cuhk.edu.hk (S.C. Liew), zsl@szu.edu.cn (S. Zhang).

1874-4907/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.phycom.2012.02.008

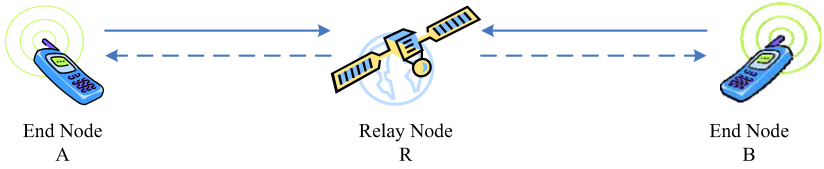


Fig. 1. System model for physical-layer network coding.

throughput of TWRC by reducing the needed time slots for the exchange of two packets from four to two [1]. In PNC, in the first time slot, end nodes *A* and *B* send signals simultaneously to relay *R*; in the second phase, relay *R* processes the superimposed signals and maps them to a network-coded packet for broadcast back to the end nodes. From the network-coded packet, each end node then makes use of its self-information to extract the packet from the other end node [1,3,4].

Prior to this paper, only a simplified version of PNC, called analog network coding (ANC) [5], has been successfully implemented. The advantage of ANC is that it is simple to implement; the disadvantage, on the other hand, is that the relay amplifies the noise along with the signal before forwarding the signal, causing error propagation.

To the best of our knowledge, the implementation of the original PNC based on XOR mapping as in [1] has not been demonstrated, even though it could have significantly better performance. A reason is that the implementation of XOR PNC poses a number of challenges. For example, the relay must be able to deal with symbol and carrier-phase asynchronies of the simultaneous signals received from the two end nodes, and the relay must perform channel estimation before detecting the signals.

This paper presents a PNC implementation in the frequency domain, referred to as FPNC, to tackle these challenges. In particular, FPNC is based on OFDM, and XOR mapping is performed on OFDM samples in each subcarrier rather than the samples in the time domain. We implement FPNC on the universal soft radio peripheral (USRP) platform. Our implementation requires only moderate modifications of the packet preamble design of 802.11 a/g OFDM PHY. With the help of the cyclic prefix (CP) in OFDM, symbol asynchrony and the multi-path fading effects can be dealt with in a similar fashion. Our experimental results show that symbol-synchronous and symbol-asynchronous FPNC have nearly the same BER performance for both channel-coded and non-channel-coded FPNC.

As far as we know, [6] is the first paper that proposes the use of OFDM for PNC. Besides discussing various issues related to the application of OFDM in PNC, it presented a theoretical analysis on applying OFDM in PNC systems to deal with the OFDM sample asynchrony problem. To tackle sample offset, the authors proposed to sample at the midpoint of the two optimal sampling points of the respective end nodes (i.e., the two adjacent peals of the matched filter output). This was considered as the best compromise. However, for real systems operating with OFDM, the sampling positions are often not important. We show in Section 2.1 that the the time-domain sampling position, in fact, has little effect on the frequency domain samples we need, since the pulse shape of the time-domain samples of OFDM is closer to a *sinc* waveform than a rectangular waveform.

Challenges

In the following, we briefly overview the challenges of PNC, and the implementation approaches taken by us to tackle them:

Asynchrony

There are two possible implementations for PNC: synchronous PNC and asynchronous PNC. In synchronous PNC, end nodes *A* and *B* have the uplink channel state information (CSI). They perform precoding and synchronize their transmissions so that their signals arrive at relay *R* with their symbols and carrier phases aligned. For high-speed transmission, such tight synchronization is challenging; in addition, timely collection of CSI is difficult in fast fading scenarios.

Asynchronous PNC is less demanding. It does not require the two end nodes to tightly synchronize and precode their transmissions. In particular, knowledge of the uplink CSI is not needed at the two end nodes. The simplicity at the end nodes comes with a cost. Without precoding and synchronization of the two end nodes, their signals may arrive at the relay with symbol and carrier-phase misalignments. A key issue in asynchronous PNC is how to deal with such signal asynchrony at the relay [7,8].

This paper focuses on the implementation of asynchronous PNC. To deal with asynchrony, our FPNC implementation makes use of OFDM to lengthen the symbol duration within each subcarrier. Then, independent XOR PNC mapping is performed within each subcarrier. OFDM splits a high-rate data stream into a number of lower-rate streams over a number of subcarriers. Thanks to the larger symbol duration within each subcarrier, the relative amount of dispersion caused by the multipath delay spread is decreased. The OFDM symbols of the two end nodes become more aligned with respect to the total symbol duration, as illustrated in Fig. 2. In particular, if the relative symbol delay is within the length of the CP, the time-domain misaligned samples will become *aligned* in the frequency domain after *DFT* is applied. This property will be elaborated later in Section 2.

Channel estimation

For good performance of asynchronous PNC, the relay must have the knowledge of the uplink CSI. This has been the assumption in many prior works on PNC (e.g., [1,6]). This means that the relay will need to estimate the channel gains. Most channel estimation techniques for the OFDM system assume point-to-point communication in which only one channel needs to be estimated. In PNC, the relay needs to estimate two channels based on simultaneous reception of signals (and preambles) from the two end nodes. This poses the following two problems in PNC that do not exist in point-to-point communication:

Please cite this article in press as: L. Lu, et al., Implementation of physical-layer network coding, Physical Communication (2012), doi:10.1016/j.phycom.2012.02.008

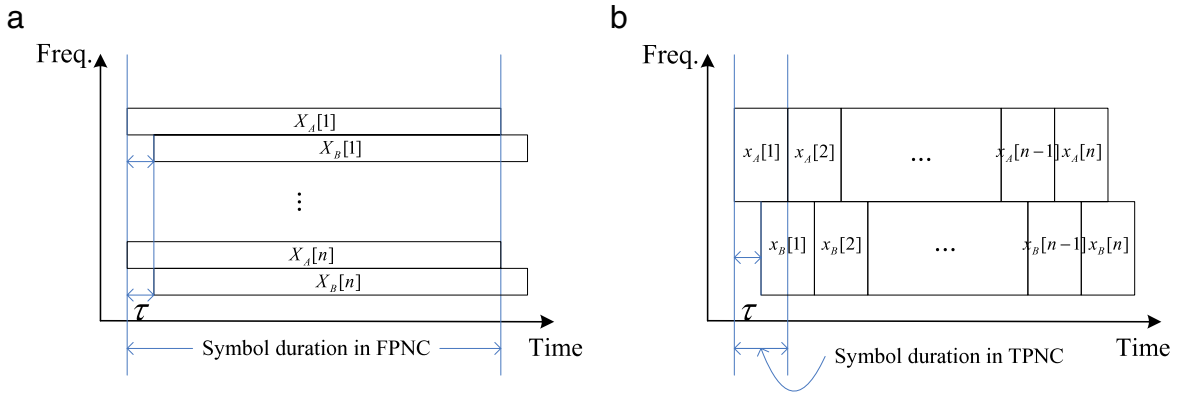


Fig. 2. PNC with time asynchrony: (a) frequency-domain physical-layer network coding (FPNC); (b) time-domain physical-layer network coding (TPNC).

- Channel estimation in a point-to-point OFDM system (e.g., 802.11 [9]) is generally facilitated by training symbols and pilots in the transmitted signal. If used unaltered in the PNC system, the training symbols and pilots from the two end nodes may overlap at the relay, complicating the task of channel estimation. In our implementation, we solve this problem by assigning orthogonal training symbols and pilots to the end nodes. The details will be given in Section 4.
- It is well known that carrier frequency offset (CFO) between the transmitter and the receiver can cause inter-subcarrier interference (ICI) if left uncorrected. In a point-to-point system, CFO can be estimated and compensated for. In PNC, we have two CFOs at the relay, one with respect to each end node. Even if the two CFOs can be estimated perfectly, their effects cannot be both compensated for totally; the total elimination of the ICI of one end node will inevitably lead to a larger ICI for the other end node. To strike a balance, our solution is to compensate for the mean of the two CFOs (i.e., compensate for $(CFO_A + CFO_B)/2$). The details will be elaborated in Section 4.

Joint channel decoding and network coding

For reliable communication in a practical PNC system, channel coding needs to be incorporated. This paper considers link-by-link channel-coded PNC, in which the relay maps the overlapped channel-coded symbols of the two end nodes [4,10] to the XOR of the source symbols[1]; after that, the relay channel-encodes the XOR source symbols to channel-coded symbols for forwarding to the end nodes. Such a link-by-link channel-coded PNC system has better performance than an end-to-end channel-coded PNC system [4,10].

In our FPNC design, we adopt the convolutional code as defined in the 802.11 a/g standard. The relay first maps the overlapped channel-coded symbols to their XOR on a symbol-by-symbol basis. After that it cleans up the noise by (i) channel-decoding the XOR channel-coded symbols to the XOR source symbols, and then (ii) re-channel-coding the XOR source symbols to the XOR channel-coded symbols for forwarding to the two end nodes.

The remainder of this paper is organized as follows: Section 2 details the delay asynchrony model of this paper. Section 3 presents the FPNC frame format design. Section 4 addresses the key implementation challenges. Experimental results are given in Section 5. Finally, Section 6 concludes this paper.

2. Effect of time-domain delay asynchrony in frequency domain

In asynchronous PNC, symbols of the two end nodes may arrive at the relay misaligned. We mentioned in the introduction that if the relative symbol delay is within the length of the CP in FPNC, then the time-domain misaligned samples will become *aligned* in the frequency domain after *DFT* is applied. This section is devoted to the mathematical derivation of this result. Here, we will derive a more general result that takes into account multi-path channels as well.

2.1. Effective discrete-time channel gains

We consider the following multi-path channel model. Suppose that there are M_A paths from node A to relay R with delays $\tau_A^0 < \tau_A^1 < \cdots < \tau_A^{M_A-1}$ and corresponding channel gains $\alpha_A^0, \alpha_A^1, \ldots, \alpha_A^{M_A-1}$. The channel impulse response of A is $g_A(t) = \sum_{i=0}^{M_A-1} \alpha_A^i \delta(t - \tau_A^i)$. Similarly, there are M_B paths from node B to relay R with delays $\tau_B^0 < \tau_B^1 < \cdots < \tau_B^{M_B-1}$ and channel gains $\alpha_B^0, \alpha_B^1, \ldots, \alpha_B^{M_B-1}$, with channel impulse response $g_B(t) = \sum_{i=0}^{M_B-1} \alpha_B^i \delta(t - \tau_B^i)$. Without loss of generality, we assume that frame A arrives earlier than frame B: specifically, $\tau_A^0 \le \tau_B^0$. *Note that our model allows for the case where nodes A and B do not exactly transmit at the same time.* If one node transmits slightly later than the other, we could simply add the lag time to all the path delays of that node. We assume that the net effect is such that the signal of A arrives earlier than the

[1] This process is called Channel-decoding-Network-Coding (CNC) in [10] because it does two things: channel decoding and network coding. Unlike the traditional multiuser detection (MUD) in which the goal is to recover the individual source information from the two end nodes, CNC aims to recover the XOR of the source information during the channel decoding process. CNC is a component in link-by-link channel-coded PNC critical for its performance [4,10].

Please cite this article in press as: L. Lu, et al., Implementation of physical-layer network coding, Physical Communication (2012), doi:10.1016/j.phycom.2012.02.008

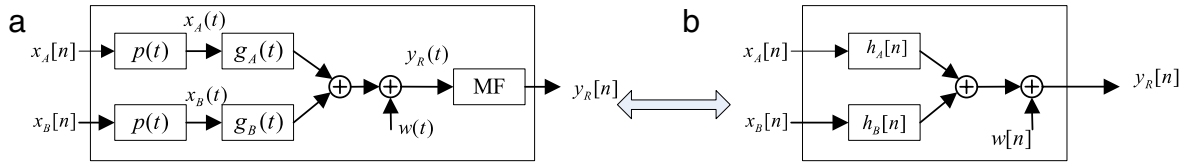


Fig. 3. (a) Continuous-time channel model for PNC, in which $x_A[n]$ and $x_B[n]$ are the time domain source samples; $y[n]$ is the time domain received samples; $g_A(t)$ and $g_B(t)$ are the wireless multipath channel gains; $p(t)$ is the pulse shaping function; $w(t)$ is the receiver noise; and MF is the matched filter and sampler at the relay node. (b) Equivalent discrete-time channel model for PNC, in which $h_A[n]$ and $h_B[n]$ denote the equivalent discrete time channel impose response (i.e., effective discrete time channel gains), and $w[n]$ is the equivalent discrete-time noise term.

signal of B, whether this is due to earlier transmission or the shorter path delay of A.

We first derive the effective discrete-time channel gains for the uplink in FPNC. As shown in Fig. 3, the discrete-time channel gains capture not just the continuous-time channel gains, but also the operations performed by pulse shaping and matched-filtering-and-sampling. Let us assume that the pulse shaping function $p(t)$ is of finite length: specifically, we assume $p(t) = 0$ for $t \le 0$ and $t \ge T_P$. The continuous-time baseband signal fed into the continuous-time channel is $x_A(t) = \sum_{n=-\infty}^{\infty} x_A[n]p(t - nT)$. The time domain received signal if only node A transmits is

$$\begin{aligned} y_A(t) &= x_A(t) * g_A(t) + w(t) \\ &= \sum_{n=-\infty}^{\infty} \sum_{i=0}^{M_A-1} \alpha_A^i x_A[n] p(t - \tau_A^i - nT) + w(t), \end{aligned} \tag{1}$$

where $w(t)$ is the noise, assumed to be AWGN. Matched-Filtering (MF) and sampling are then performed on (1), by sampling at the first multipath channel tap of the uplink channel between node A and relay R, to get the received samples

$$\begin{aligned} y_A[m] &= \int_{-\infty}^{\infty} y(t) p(t - \tau_A^0 + T_P - mT) dt \\ &= \sum_{n=-\infty}^{\infty} x_A[n] \left\{ \int_{-\infty}^{\infty} \sum_{i=0}^{M_A-1} \alpha_A^i p(t - \tau_A^i - nT) \right. \\ &\quad \left. \times\ p(t - \tau_A^0 + T_P - mT) dt \right\} + w[m] \\ &= \sum_{n=-\infty}^{\infty} x_A[n] h_A[m - n] + w[m], \end{aligned} \tag{2}$$

where $w[m] = \int_{-\infty}^{\infty} w(t) p(t - \tau_A^0 + T_P - mT) dt$. We see that the effective discrete-time channel of A is such that $h_A[m - n] = \int_{-\infty}^{\infty} \sum_{i=0}^{M_A-1} \alpha_A^i p(t - \tau_A^i - nT)\, p(t - \tau_A^1 + T_P - mT) dt$.

Note that $p(t - \tau_A^i - nT) p(t - \tau_A^0 + T_P - mT) = 0$ if $|\tau_A^i - \tau_A^0 + T_P - (m-n)T| \ge T_P$. In other words, $h_A[m-n] = 0$ for $(m-n)T \ge \tau_A^{M_A-1} - \tau_A^0 + 2T_P$ and $(m-n)T \le 0$. Define $D_A = \lceil (\tau_A^{M_A-1} - \tau_A^0 + 2T_P)/T \rceil$.

Let us now consider what if both end nodes transmit. The received signal at the relay node is

$$y_R(t) = x_A(t) * g_A(t) + x_B(t) * g_B(t) + w(t). \tag{3}$$

Sticking to the above MF that is defined with respect to the first path delay of A, we have

$$y_R[n] = x_A[n] * h_A[n] + x_B[n] * h_B[n] + w[n], \tag{4}$$

where $h_A[n] = 0$ for $n < 0$ and $n \ge D_A \triangleq \lceil (\tau_A^{M_A-1} - \tau_A^0 + 2T_P)/T \rceil$, and $h_B[n] = 0$ for $n < \lceil (\tau_B^0 - \tau_A^0 + 2T_P)/T \rceil$ and $n \ge D_B \triangleq \lceil (\tau_B^{M_B-1} - \tau_A^0 + 2T_P)/T \rceil$.

2.2. Delay-spread-within-CP requirement

The delay spread of node A is D_A, and the delay spread of node B, with respect to time $n = 0$, is D_B. We define the delay spread of the PNC system as (i.e., it combines the delay spreads of A and B into a potentially larger delay spread, as illustrated in Fig. 4)

$$\text{delay spread} = \max[D_A, D_B]. \tag{5}$$

The above derivation is general and does not have any requirement on the modulation. In this subsection, we will present the OFDM modulated PNC system. In particular, we will present the "Delay-Spread-Within-CP Requirement" for FPNC. That is, we combine the Cyclic Prefix (CP) and Discrete Fourier Transform (*DFT*) to show that the time-domain symbol asynchrony of FPNC disappears in the frequency domain, when the uplink frames satisfy the *Delay-Spread-Within-CP* Requirement.

First, let $H[k]$ be the N-point *DFT* of $h[n]$ given by Oppenheim and Schafer [11] (Note that, in the following derivation, we assume the subcarrier indices start from 0, i.e., $k = 0, \ldots, N-1$)

$$H[k] = DFT\{h[n]\} = \sum_{n=0}^{N-1} h[n] e^{-j\frac{2\pi nk}{N}}, \quad 0 \le k \le N-1. \tag{6}$$

The N-point circular convolution of x_n and h_n is written as

$$y[n] = x[n] \otimes_N h[n] = \sum_{k=0}^{N-1} h[k] x[n-k]_N, \tag{7}$$

where $[n-k]_N$ denotes $[n-k]$ modulo N. In other words, $x[n-k]_N$ is a periodic version of $x[n-k]$ with period N. From the definition of *DFT*, circular convolution in time leads to multiplication in the frequency [11]:

$$DFT\{x[n] \otimes_N h[n]\} = X[k]H[k], \quad 0 \le k \le N-1. \tag{8}$$

The channel output, as in (4), however, is not a circular convolution but a linear convolution. *The linear convolution between the channel input and impulse response can be*

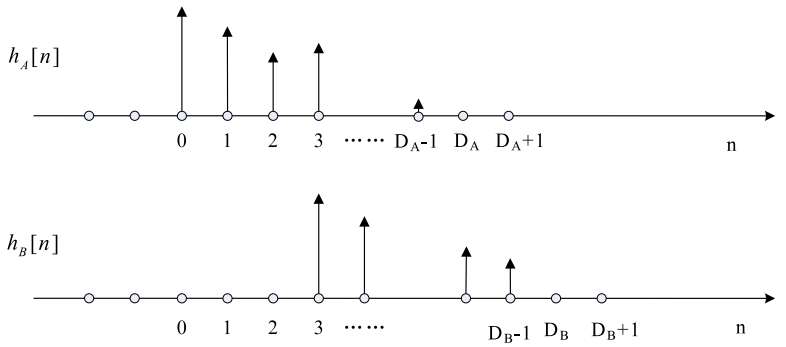


Fig. 4. Example of delay spread in FPNC.

turned into a circular convolution by adding a special prefix to the input called a cyclic prefix (CP) [12].

For FPNC, let $H_A[k]$ and $H_B[k]$ denote the frequency responses of the discrete-time channels, and let C denote the length of the CP. One OFDM symbol duration is then $N+C$. The CP for $x_A[n]$ is defined as $x_A[N-C], \ldots, x_A[N-1]$: it consists of the last C values of the $x_A[n]$ sequence. For each input sequence of length N, these last C samples are appended to the beginning of the sequence. This yields a new sequence $x_A^{OFDM}[n]$, $-C \le n \le N-1$, of length $N+C$, where $x_A^{OFDM}[-C], \ldots, x_A^{OFDM}[N-1] = x_A[N-C], \ldots, x_A[N-1], x_A[0], \ldots, x_A[N-1]$. Note that with this definition, $x_A^{OFDM}[n] = x_A[n]_N$ for $-C \le n \le N-1$, which implies that $x_A^{OFDM}[n-k] = x_A[n-k]_N$ for $-C \le n-k \le N-1$.

Suppose $x_A^{OFDM}[n]$ and $x_B^{OFDM}[n]$ are inputs to a discrete-time channel with impulse response $h_A[n]$ and $h_B[n]$, respectively. The channel output $y_R[n]$, $0 \le n \le N-1$ is then (assuming that the delay spread of FPNC $\max[D_A, D_B]$ is no larger than the CP length C)

$$\begin{aligned} y_R[n] &= x_A^{OFDM}[n] * h_A[n] + x_B^{OFDM}[n] * h_B[n] + w[n] \\ &= \sum_{k=0}^{C-1} h_A[k] x_A^{OFDM}[n-k] \\ &\quad + \sum_{k=0}^{C-1} h_B[k] x_B^{OFDM}[n-k] + w[n] \\ &= \sum_{k=0}^{C-1} h_A[k] x_A[n-k]_N \\ &\quad + \sum_{k=0}^{C-1} h_B[k] x_B[n-k]_N + w[n] \\ &= x_A[n] \otimes_N h_A[n] + x_B[n] \otimes_N h_B[n] + w[n], \end{aligned} \tag{9}$$

where the third equality follows from the fact that for $0 \le k \le C-1$, $x_A^{OFDM}[n-k] = x_A[n-k]_N$ for $-C \le n-k \le N-1$. Thus, by appending a CP to the channel input, the linear convolution associated with the channel impulse response $y_R[n]$ for $0 \le n \le N-1$ becomes a circular convolution. Taking the *DFT* of the channel output in the absence of noise then yields the following FPNC frequency domain digital expression:

FPNC *frequency domain digital expression*:

Define C as the length of the CP, and assuming FPNC delay spread $= \max[D_A, D_B] \le C$ (where D_A and D_B are functions of the multipath delays $\tau_A^{M_A-1}$ and $\tau_B^{M_B-1}$, respectively) the received signal at subcarrier k is given by

$$\begin{aligned} Y[k] &= H_A[k]X_A[k] + H_B[k]X_B[k] + W[k], \\ &\quad k = 0, \ldots, N-1. \end{aligned} \tag{10}$$

Note that the time-domain delay spread has been incorporated into $H_A[k]$ and $H_B[k]$ respectively. In FPNC, we will map $Y[k]$ for each subcarrier k into the XOR, $X_A[k] \oplus X_B[k]$. This will be detailed in Section 4.3. The main point here is, in (10), the signals of different subcarriers k are isolated from each other, and we only need to perform PNC mapping within each subcarrier.

We remark that our discussion so far in this section has assumed the absence of CFO. When there is CFO, inter-carrier interference (ICI) may occur, and this will be further discussed in Section 4.1.

3. FPNC frame format

This section focuses on the PHY frame design to enable asynchronous operation, channel estimation, and frequency offset compensation in FPNC. As previously mentioned, the asynchronous operation requires the PNC delay spread to be within CP. A simple MAC protocol as follows could be used to trigger near-simultaneous transmissions by the two end nodes to ensure this. The relay could send a short polling frame (similar to the "beacon frame" in 802.11 that contains only 10 bytes) to the end nodes. Upon receiving the polling frame, the end nodes then transmit. With this method, the symbols would arrive at the relay with a relative delay offset of $|RTT_A - RTT_B|$, where RTT is the round trip time, including the propagation delay and the processing time at the end nodes. This delay offset is not harmful to our system as long as the sample misalignment of two end nodes is within the CP length. In our experiments, instead of relying on the polling MAC protocol above, we use a different mechanism to control the relative delay offset so that we could systematically try out different offsets and see their effects. Details can be found in Section 5.2.

Given this loose synchronization, our training symbols and pilot designs described below can then be used to facilitate channel estimation and frequency offset compensation in FPNC. We modify the PHY preamble

Please cite this article in press as: L. Lu, et al., Implementation of physical-layer network coding, Physical Communication (2012), doi:10.1016/j.phycom.2012.02.008

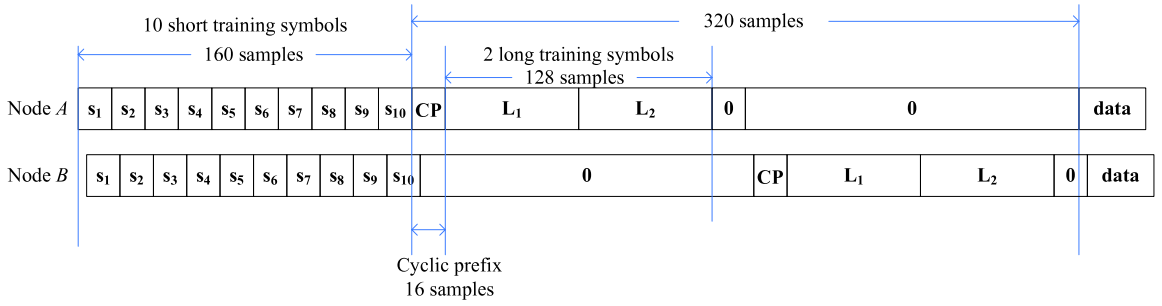


Fig. 5. FPNC preamble format.

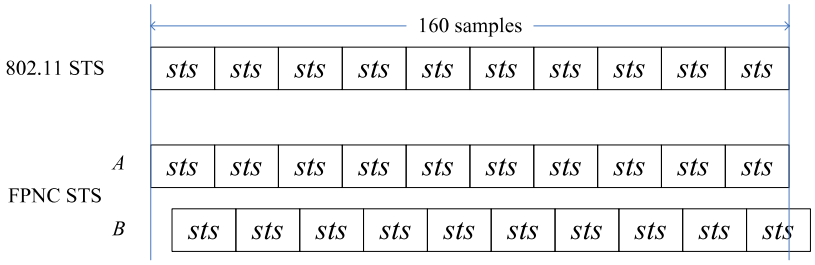


Fig. 6. Short training symbol design for FPNC (time domain).

design of 802.11 a/g for FPNC. The overall FPNC frame format is shown in Fig. 5. The functions of the different components in the PHY preamble are described in the next few subsections.

3.1. FPNC short training symbol

In 802.11, the short training symbol (STS) sequence contains 160 time-domain samples, in which 16 samples form one STS unit (*sts*) for a total of 10 identical units, as shown in Fig. 6. FPNC adopts the same STS sequence as in 802.11, as illustrated in Fig. 6. The STS sequence is used by the relay node to perform the sample timing recovery on the received frame. In particular, the relay node applies a cross-correlation to locate the sample boundary for the long training symbols that follow the STS sequence. The normalized cross-correlation is defined as follows:

$$Z[n] = \frac{\left|\sum_{i=0}^{L-1}(sts^*[i]y_R[n+i])\right|}{\sum_{i=0}^{L-1}(y_R[n+i]y_R^*[n+i])}, \tag{11}$$

where n is the received sample index, $y_R[n]$ is the n-th sample at the relay R, and $L = 16$ is the length of each *sts*. For FPNC, this cross-correlation will result in 20 peaks over the STS sequences (see Fig. 7) of the two frames if the frames are not synchronized. From this profile of peaks, we can identify the last two peaks. If the *Delay-Spread-Within-CP* requirement is satisfied, then the last two peaks must be the last peaks of *A* and *B*, respectively. This is because the CP as well as the *sts* are of 16 samples in length. From there, we could locate the boundaries of the long training symbol (*LTS*) of *A* and *B* that follow. Note that when the STS sequences of nodes *A* and *B* overlap exactly, we will have ten peaks only. In this case, the *LTS* boundaries of *A* and *B* also overlap exactly, and we simply use the last peak to identify the common boundary.

3.2. FPNC long training symbol

With reference to Fig. 8, the 802.11 *LTS* sequence contains 160 time-domain samples in which there is a CP followed by two identical *LTS* units, *lts*. The receiver uses the *LTS* sequence to perform channel estimation and CFO compensation.

For FPNC, in order to estimate two uplink channel gains, we design the *LTS* so that it contains twice the length of *LTS* in 802.11 a/g, as shown in Fig. 8. In Fig. 8, we intentionally show the case in which the *LTS* sequences of the two end nodes are not exactly synchronized. Note that we change the 802.11 *LTS* design by shortening its original CP length from 32 to 16 to make sure that the two *lts* units of *B* will not overlap with the data of *A* that follows under the condition that the delay spread is less than the CP length of 16. This does not impose additional requirement on the delay spread, since the CP of the data OFDM symbols in 802.11 a/g (and FPNC) have only 16 samples anyway (i.e., the delay spread must be within 16 samples anyway). Section 4 will detail the CFO compensation and channel estimation methods for our implementation.

3.3. FPNC pilot

There are four pilots for each OFDM symbol in 802.11, as shown in Fig. 9. The four pilots are used to fine-tune the channel gains estimated from *LTS*. In a frame, there are multiple OFDM symbols, but only one *LTS* in the beginning. In practice, the channel condition may have changed by the time the later OFDM symbols arrive at the receiver. That is, the original channel gains as estimated by *LTS* may not be

Please cite this article in press as: L. Lu, et al., Implementation of physical-layer network coding, Physical Communication (2012), doi:10.1016/j.phycom.2012.02.008

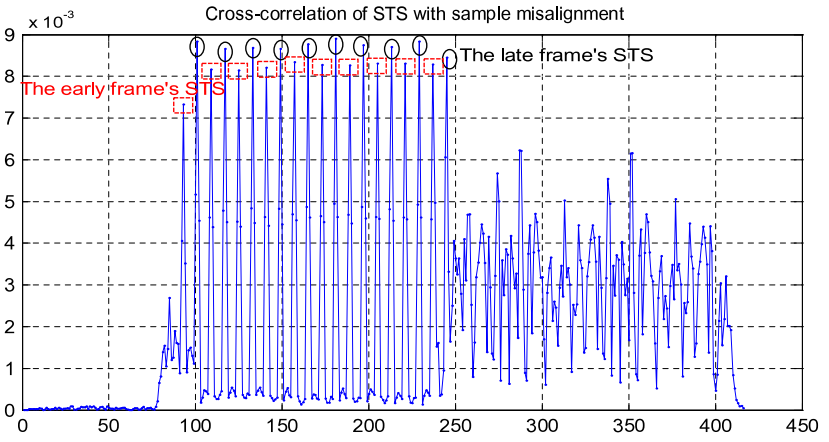


Fig. 7. Cross-correlation of the STS for the uplink of FPNC.

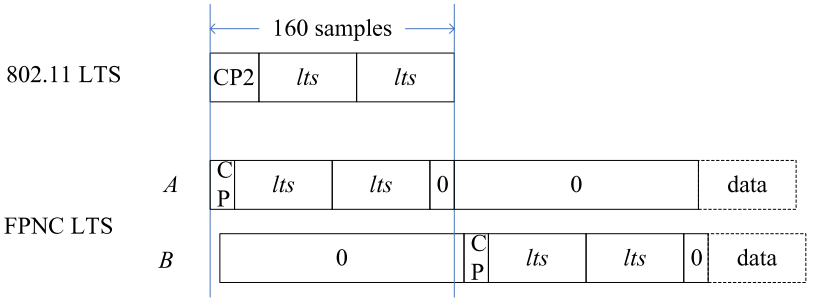


Fig. 8. Long training symbol design for FPNC (time domain).

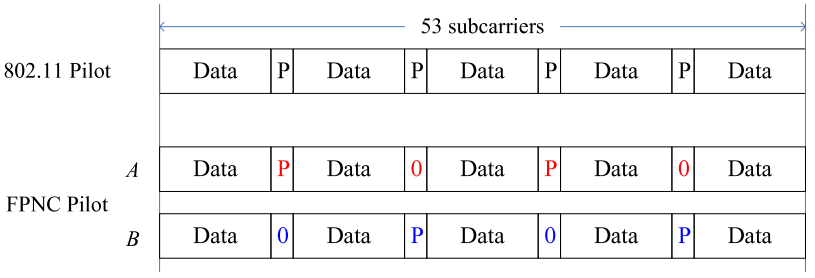


Fig. 9. Long training symbol design for FPNC (time domain).

accurate anymore for the later OFDM symbols. The pilots are used to track such channel changes.

In FPNC, we design the FPNC pilots of nodes *A* and *B* by nulling certain pilots to introduce orthogonality between them, as shown in Fig. 9. As will be detailed in Section 4.2, this allows us to track the channel gains of *A* and *B* separately in a disjoint manner in FPNC. We conducted some experiments for a point-to-point communication system using the two-pilot design rather than the four-pilot design. We find that for our linear interpolation channel tracking scheme described in Section 4.2, the BER performances of the two-pilot and four-pilot designs are comparable for BPSK- and QPSK-modulated systems.

4. Addressing key implementation challenges in FPNC

We next present our methods for carrier frequency offset compensation, channel estimation, and FPNC mapping, assuming the use of the PHY frame format presented in Section 3.

4.1. FPNC carrier frequency offset (CFO) compensation

For CFO compensation, we first estimate the two independent CFOs (namely CFO_A and CFO_B) caused by the carrier frequency offsets between nodes *A* and *B* and relay *R*, respectively. We then compensate for the mean of the two CFOs (i.e., $CFO_{PNC} = (CFO_A + CFO_B)/2$). The details are presented below.

4.1.1. CFO estimation

For the uplink phase, when there are CFOs, the received frames at relay R will suffer from time-varying phase asynchronies. We need to compensate for the CFOs to

alleviate inter-carrier interference (ICI) among data on different subcarriers.

Recall that in Section 3, we mentioned that a loose synchronization MAC protocol can be used to ensure that the difference of the arrival times of the frames from nodes A and B are within CP. That means that the LTSs from nodes A and B will overlap with each other substantially, with the non-overlapping part smaller than CP (see Fig. 5). Recall also that we introduce orthogonality between the LTSs of nodes A and B so that when the LTS units in A are active, the LTS in B are zeros, and vice versa, as shown in Fig. 8. This allows us to separately estimate CFO_A and CFO_B. Without loss of generality, in the following we focus on the estimation of CFO_A using LTS_A.

CFO_A is given by $\Delta f_A = f_A - f_R$ (i.e., the difference in the frequencies of the oscillators of node A and relay R). We define the normalized CFO_A to be $\phi_A = 2\pi\,\Delta f_A \frac{T}{N}$, where T is the duration of one OFDM symbol, and N is the number of samples in one OFDM symbol not including CP. In other words, ϕ_A is the additional phase advance introduced by the CFO from one sample to the next sample.

To estimate ϕ_A, we multiply one sample in the first unit of LTS_A (see Fig. 8) by the corresponding sample in the second unit of LTS_A to obtain $(y_R^{LTS_A}[n])^* y_R^{LTS_A}[n+N]$. Then, $angle((y_R^{LTS_A}[n])^* y_R^{LTS_A}[n+N]) \in (-\pi, \pi)$ is given by

$$angle((y_R^{LTS_A}[n])^* y_R^{LTS_A}[n+N]) + 2m\pi = N\phi_A, \tag{12}$$

where $m \in \{\ldots, -2, -1, 0, 1, 2, \ldots\}$.

For our experimental test-bed, USRP, we found that the accuracies of the onboard oscillators are such that they do not induce large CFOs so that $m = 0$ (interested readers are referred to [13] for CFO estimation when $m \neq 0$). Hence, we could write (12) as follows:

$$N\phi_A = angle((y_R^{LTS_A}[n])^* y_R^{LTS_A}[n+N]). \tag{13}$$

Strictly speaking, (13) is an expression for the noiseless case. Because of noise, $angle((y_R^{LTS_A}[n])^* y_R^{LTS_A}[n+N])$ for different $n \in \{0, \ldots, N-1\}$ could be different. Thus, in our computation, we first obtained $\hat{\phi}_A[n] = angle((y_R^{LTS_A}[n])^* y_R^{LTS_A}[n+N])$ for $n = 0, \ldots, N-1$, and then estimate ϕ_A by

$$\hat{\phi}_A = \underset{n\in\{0,\ldots,N-1\}}{median}(\hat{\phi}_A[n]). \tag{14}$$

We obtain ϕ_B similarly.

The reason we use the median CFO instead of the mean values is that we find the median is more stable. In particular, some samples of $\hat{\phi}_A[n]$ are outliers that appear to be caused by unknown errors of significant magnitudes. We will show the BER results comparing the use of mean and median for CFO compensation (in Fig. 10(b)).

4.1.2. Compensation for two CFOs

In FPNC, we adopt the mean of the two CFOs for compensation purposes:

$$\tilde{\phi} = (\hat{\phi}_A + \hat{\phi}_B)/2. \tag{15}$$

Experimental results show (see Fig. 10(a)) that compensation by the mean $\tilde{\phi}$ in (15) is better than compensation by either $\hat{\phi}_A$ or $\hat{\phi}_B$. We believe a theoretical study to explore and compare different compensatation methods may be worthwhile in the future. As far as we know, there have been no theoretical treatments of compensating for two CFOs. Fig. 10(b) shows the BER performance of using the median for the estimate of $\hat{\phi}_A$ or $\hat{\phi}_B$ as in (14), versus using the mean. It shows that the use of the median results in better performance.

After compensation, our received data in the time domain is given by

$$\tilde{y}_R[n] = y_R[n]e^{-jn\tilde{\phi}}. \tag{16}$$

In the frequency domain, we have

$$\tilde{Y}_R[k] = DFT(\tilde{y}_R[n]). \tag{17}$$

We should emphasize that the computation complexity of FPNC CFO compensation is exactly the same as that of point-to-point communication, thus real-time decoding is possible.

4.2. FPNC channel estimation

In this subsection, we present the channel estimation and tracking method for FPNC. Note that CFO compensation was performed on the time-domain signal. For channel estimation, however, we are interested in the channel gains for different subcarriers in the frequency domain. This means that channel estimation will be performed after DFT. Thus, in the following we look at the signal after CP removal and DFT.

For FPNC channel estimation, we use the LTS to obtain a first estimate. Pilots are used to obtain additional estimates for later OFDM symbols within the same frame. In the following, we consider channel estimation of $H_A[k]$. Estimation of $H_B[k]$ is performed similarly.

For channel estimation based on LTS, define one FPNC LTS unit of node A (i.e., with respect to Fig. 8, one unit is lts_A) in the frequency domain as $X_A^{LTS}[k]$, where $k = 0, \ldots, N-1$. Based on the first unit of lts_A the received frequency domain LTS_A (i.e., $\tilde{Y}_R^{LTS_A}[k] = DFT(\tilde{y}_R^{LTS_A}[n])$), we perform channel estimation of $H_A[k]$ as follows:

$$\hat{H}_A[k] = \frac{\tilde{Y}_R^{LTS_A}[k]}{X_A^{LTS}[k]}. \tag{18}$$

As mentioned in Section 3, each LTS contains two identical units in our design. The uplink channel gain $H_A[k]$ between node A and relay R is estimated by taking the average of the two units results

$$\tilde{H}_A[k] = (\hat{H}_A[k] + \hat{H}_A[k+N])/2. \tag{19}$$

In general, the channel may have changed from the first OFDM symbol to the last OFDM symbol within the same frame. The estimate based on LTS in (19) applies only for the earlier symbols. Pilots are used to track the channel changes for later symbols. Our pilot design was shown in Fig. 9. In each FPNC OFDM symbol, there are two pilots per

Please cite this article in press as: L. Lu, et al., Implementation of physical-layer network coding, Physical Communication (2012), doi:10.1016/j.phycom.2012.02.008

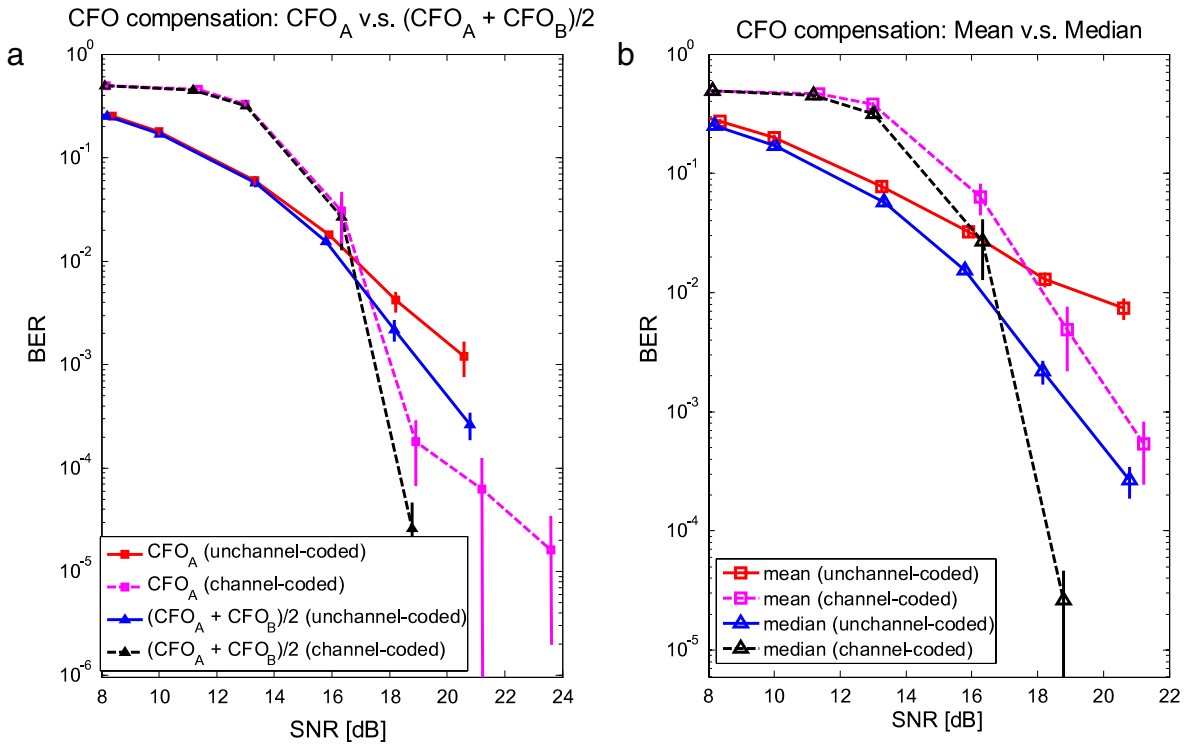


Fig. 10. Comparison of the different CFO compensation methods: (a) CFO_A vs. $\frac{CFO_A+CFO_B}{2}$; (b) mean CFO vs. median CFO.

end node. Note from Fig. 9 that the two pilots of node A and the two pilots of node B are positioned at different subcarriers and non-overlapping in the frequency domain. Therefore, we could separately track the changes in $H_A[k]$ and $H_B[k]$. In the following, we consider the tracking of $H_A[k]$. Tracking of $H_B[k]$ can be done similarly.

Let k' and k'' denote the subcarriers occupied by the two pilots of A. Consider the OFDM symbol m. Let $\tilde{Y}_R^m[k']$ and $\tilde{Y}_R^m[k'']$ be the received signal in the frequency domain. Because the pilots of A and B do not overlap, $\tilde{Y}_R^m[k']$ and $\tilde{Y}_R^m[k'']$ contain only signals related to the pilots of A. We first multiply $\tilde{Y}_R^m[k']$ and $\tilde{Y}_R^m[k'']$ by $(\tilde{H}_A^m[k'])^{-1}$ and $(\tilde{H}_A^m[k''])^{-1}$ obtained from (19), respectively. Let $P_A[k']$ and $P_A[k'']$ be the two pilots. Then, we compute

$$\begin{aligned}\Delta\tilde{H}_A^m[k'] &= (\tilde{H}_A^m[k'])^{-1}\tilde{Y}_R^m[k']/P_A[k'],\\ \Delta\tilde{H}_A^m[k''] &= (\tilde{H}_A^m[k''])^{-1}\tilde{Y}_R^m[k'']/P_A[k''].\end{aligned} \tag{20}$$

After that, we perform linear fitting to obtain $\Delta\tilde{H}_A^m[k]$ for $k \neq k', k''$, as follows:

$$\Delta\tilde{H}_A^m[k] = \Delta\tilde{H}_A^m[k'] + \left(\frac{\Delta\tilde{H}_A^m[k''] - \Delta\tilde{H}_A^m[k']}{k'' - k'}\right)(k - k'). \tag{21}$$

To obtain the final channel estimation for the m-th OFDM symbol, we compute

$$H_A^m[k] = \tilde{H}_A[k] \cdot \Delta\tilde{H}_A^m[k]. \tag{22}$$

4.3. FPNC mapping

For reliable communication, channel coding should be used. Channel coding in PNC systems can be either done on an end-to-end basis or a link-by-link basis [10,4]. The latter generally has better performance because the relay performs channel decoding to remove noise before forwarding the network-coded signal.

The basic idea in link-by-link channel-coded PNC is shown in Fig. 11. It consists of two parts. Let $\bar{Y}_R$ denote the vector representing the overall channel-coded overlapped frames received by relay R. The operation performed by the first part is referred to as the Channel-decoding and Network-Coding (CNC) process in [10]. It maps $\bar{Y}_R$ to $\bar{S}_A \oplus \bar{S}_B$, where $\bar{S}_A$ and $\bar{S}_B$ are the vectors of source symbols from nodes A and B, respectively, and the $\oplus$ operation represents symbol-by-symbol XOR operation across corresponding symbols in $\bar{S}_A$ and $\bar{S}_B$. Note that the number of symbols in $\bar{Y}_R$ is more than the number of symbols in $\bar{S}_A \oplus \bar{S}_B$ because of channel coding. Importantly, CNC involves both channel decoding and network coding. In particular, CNC channel-decodes the received signal $\bar{Y}_R$ not to $\bar{S}_A$ and $\bar{S}_B$ individually, but to their XOR. The second part can be just any conventional channel coding operation that channel code $\bar{S}_A \oplus \bar{S}_B$ to $\bar{X}_R = C(\bar{S}_A \oplus \bar{S}_B)$ for broadcast to nodes A and B, where $C(*)$ is the channel coding operation.

As mentioned in [10,4], the CNC component is unique to the PNC system, and different designs can have different performance and different implementation complexity. We refer the interested readers to [4] for a discussion on different CNC designs.

Please cite this article in press as: L. Lu, et al., Implementation of physical-layer network coding, Physical Communication (2012), doi:10.1016/j.phycom.2012.02.008

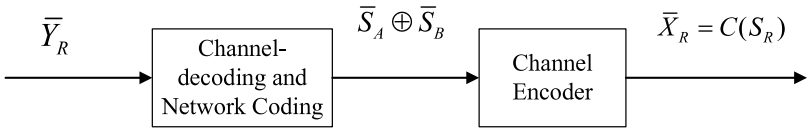


Fig. 11. Link-by-link channel-coded PNC, including channel-decoding and network coding (CNC) process and channel encoding.

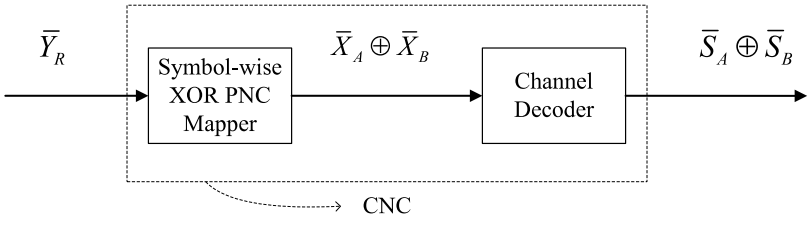


Fig. 12. XOR-CD design for CNC.

In this paper, we choose a design that is amenable to simple implementation, as shown in Fig. 12. We refer to this CNC design as XOR-CD. In this design, any linear channel code can be used. In our implementation, we choose to use the convolutional code. In XOR-CD, the channel-decoding and network coding operations in CNC are performed in a disjoint manner. As shown in Fig. 12, based on the CFO-compensated $\tilde{Y}_R[k]$ obtained as in (17), we obtain the overall vector $\bar{Y}_R = (Y_R[k])_{k=0,1,\ldots}$. We then perform symbol-wise PNC mapping to get an estimate for the channel-coded XOR vector $\bar{X}_A \oplus \bar{X}_B = (X_A[k] \oplus X_B[k])_{k=0,1,\ldots}$, where $\bar{X}_A = (X_A[k])_{k=0,1,\ldots}$ and $\bar{X}_B = (X_B[k])_{k=0,1,\ldots}$ are the channel-coded vectors from A and B, respectively. We assume the same linear channel code is used at nodes A, B, and R. Note that since we adopt the convolutional code, $C(*)$ is linear. Therefore, we have $\bar{X}_A \oplus \bar{X}_B = C(\bar{S}_A) \oplus C(\bar{S}_B) = C(\bar{S}_A \oplus \bar{S}_B)$, and thus the same Viterbi channel decoder as used in a conventional point-to-point communication link can be used in the second block of Fig. 12.

The mapping in the first block in Fig. 12 could be performed as follows. Based on the channel gains estimated in (22), we could perform the XOR mapping for the k-th subcarrier in the m-th OFDM symbol (assuming BPSK modulation) according to the decision rule below:

$$\begin{aligned}
&\exp\left\{-\frac{|Y_R^m[k]-H_A^m[k]-H_B^m[k]|^2}{2\sigma^2}\right\}\\
&\quad+\exp\left\{-\frac{|Y_R^m[k]+H_A^m[k]+H_B^m[k]|^2}{2\sigma^2}\right\}\\
&\underset{X_R^m[k]=1}{\overset{X_R^m[k]=-1}{\gtrless}}\exp\left\{-\frac{|Y_R^m[k]+H_A^m[k]-H_B^m[k]|^2}{2\sigma^2}\right\}\\
&\quad+\exp\left\{-\frac{|Y_R^m[k]-H_A^m[k]+H_B^m[k]|^2}{2\sigma^2}\right\},
\end{aligned}\tag{23}$$

where we have assumed Gaussian noise with variance σ^2. The computation complexity in (23),[2] however, is large. In our implementation, we adopt a simple "log-max approximation" [14] (i.e., $\log(\sum_i \exp(z_i)) \approx \max_i z_i$) that yields the following decision rule:

$$\begin{aligned}
&\min\{|Y_R^m[k]-H_A^m[k]-H_B^m[k]|^2,\\
&\quad|Y_R^m[k]+H_A^m[k]+H_B^m[k]|^2\}\\
&\underset{X_R^m[k]=-1}{\overset{X_R^m[k]=1}{\gtrless}}\min\{|Y_R^m[k]+H_A^m[k]-H_B^k[k]|^2,\\
&\quad|Y_R^m[k]-H_A^m[k]+H_B^m[k]|^2\}.
\end{aligned}\tag{24}$$

This decision rule can also be interpreted as in Table 1, where

$$U \overset{\Delta}{=} \underset{U\in\{\pm H_A^m[k]\pm H_B^m[k]\}}{\arg}\ \min\{|Y_R^m[k]-U|^2\}.\tag{25}$$

Note here that this decision rule could be used even for non-Gaussian noise. This is because (25) corresponds to finding the nearest point in the constellation map (constructed by combining the two end nodes' channel gains).

Based on the XORed samples detected using the decision rule of Table 1, we then perform the channel decoding to get the XORed source samples. In our implementation, we use a Viterbi decoder with hard input and hard output. In general, a soft Viterbi algorithm could also be used for potentially better BER performance [15].

Table 1
XOR mapping with BPSK modulation in FPNC.

$U = \arg_{U\in\{\pm H_A^m[k]\pm H_B^m[k]\}} \min\{\lvert Y_R^m[k]-U\rvert^2\}$	$X_R^m[k] = X_A^m[k] \oplus X_B^m[k]$
$H_A^m[k]+H_B^m[k]$	1
$H_A^m[k]-H_B^m[k]$	−1
$-H_A^m[k]+H_B^m[k]$	−1
$-H_A^m[k]-H_B^m[k]$	1

5. Experimental results

This section presents details of our FPNC implementation over the software radio platform and the experimental results.

[2] Note that (23) is similar to (7) in Ref. [8], except that here we allow for the possibility that $|H_A^m[k]| \neq |H_B^m[k]|$.

5.1. FPNC implementation over software radio platform

We implement FPNC in a 3-node GNU Radio testbed, with Software Defined Radio (SDR). The topology is shown in Fig. 1. Each node is a commodity PC connected to a USRP GNU radio [16].

- **Hardware:** We use the Universal Software Radio Peripheral (USRP) [17] as our radio hardware. Specifically, we use the XCVR2450 daughterboard operating in the 2.4/5 GHz range as our RF frontend. We use the USRP1 motherboard for baseband data processing. The largest bandwidth that USRP1 could support is 8 MHz. In our experiment, we use only half of the total bandwidth for FPNC (i.e., 4 MHz bandwidth).
- **Software:** The software for baseband signal processing is based on the open source of the GNURadio project [16]. We build our system by modifying the 802.11 g transmitter implementation in the FTW project [18]. The FTW project [19], however, does not have a 802.11 g receiver. Therefore, we develop our own OFDM receiver, designed specifically to tackle various issues in the FPNC system, such as CFO estimation and compensation, channel estimation, and CNC processing as presented in Section 4.

5.2. Experimental results

We conduct our experiments over channel one of 802.11 g, with 2.412 GHz being the central frequency. For each transmitter power level (we vary the SNR from 5 dB to 20 dB), we transmit 1000 packets and examine the resulting BER performance. Both the symbol-synchronous and symbol-asynchronous cases are investigated. The packet length is 1500 bytes, which is a normal Ethernet frame size.

5.2.1. Time-synchronous FPNC versus time-asynchronous FPNC

In Section 2, we derived theoretically that as long as the *Delay-Spread-Within-CP* requirement is satisfied, FPNC will not have asynchrony in the frequency domain. Of interest is whether this reduces the asynchrony penalty in practice. In our first set of experiments, we investigate this issue. We study both non-channel-coded as well as channel-coded FPNC systems.

To establish a CP level synchronization between two end nodes, we use the MIMO transmitter setup for USRP in our experiment. In particular, when using the USPR1s as the end nodes, we plug two daughterboards (i.e. two end nodes) into one motherboard so that they have the same reference clock. We also set the MUX on the FPGA to let the two daughterboards transmit simultaneously. The antennas of the two daughterboards are positioned further from each other through cables.

We simulate different levels of asynchrony in a controlled manner by inserting zeros for one transmission path (i.e., silence period at the beginning) and adjusting the distance between the antennas of the end nodes (i.e., to purposely introduce delay to one transmission path.). One of the setups corresponds to the perfectly synchronized case (the STS correlation has only ten peaks in the perfectly synchronized case: see Section 3). Fig. 13(a) shows the BER-SNR curves for the synchronous case, and Fig. 13(b) shows the curves for the asynchronous case with eight samples offset between the early and late frames. Note that this asynchrony still satisfies the *Delay-Spread-Within-CP* requirement because the CP has 16 samples. We find that the performance results of the asynchronous cases with other time offsets to be similar, and we therefore present the results of the eight-sample offset only.

The above controlled experiment verifies that if we can synchronize the simultaneous transmissions to within CP, then the OFDM PNC system will work. We are currently working on implementing a MAC protocol that can ensure within-CP synchronization. We believe coordination through beacons from the relay will allow us to achieve that.

From Fig. 13(a) and (b), we see that the asynchronous FPNC has essentially the same BER performance as that of the synchronous FPNC. Hence, we conclude that FPNC is robust against time asynchrony as far as BER performance is concerned. To better analyze the BER performance of FPNC, we benchmark it with a standard point-to-point transmission, in which only node *A* communicates with relay *R*. It is shown in [20] that the non-channel-coded BER performance loss for FPNC is 4–5 dB, while the channel-coded BER performance loss is less than 3 dB, relative to point-to-point transmission. If we use the guideline that the common decodable 802.11 link usually works at an SNR regime that is higher than 20 dB [5,21], we can conclude that our FPNC implementation has very good performance in this regime (with BER lower than 10^{-5}) that it could nearly double the throughput of a TWRC compared to the traditional scheduling method (i.e., FPNC reduces the needed time slots for TWRC from four to two).

5.2.2. FPNC versus other approaches for TWRC

Our next set of experiments is geared toward the comparison of FPNC with other TWRC schemes. Recall that FPNC TWRC is a two-phase scheme using two time slots for the exchange of a pair of packets between two end nodes. We consider the following two additional approaches [4]

- **SNC:** The straightforward network coding (SNC) scheme makes use of conventional network coding at the higher layer using three time slots. In SNC, node *A* transmits to relay *R* in the first time slot; node *B* transmits to relay *R* in the second time slot; relay *R* then XOR the two packets from *A* and *B* and transmits the XOR packets to nodes *A* and *B* in the third time slot.
- **TS:** Traditional scheduling (TS) scheme uses four time slots. In the first time slot, node *A* transmits to relay *R*; in the second time slot, relay *R* forwards the packet from *A* to node *B*. Similarly, the packet from node *B* to node *A* uses two additional time slots for its delivery.

Our overall goal is to compare the throughputs of the three schemes. To derive the throughputs, we first measure the following three frame-error rates:

(1) $FER_{PNC} = P_f^{uplink,PNC}$: frame-error rate of the uplink of FPNC.

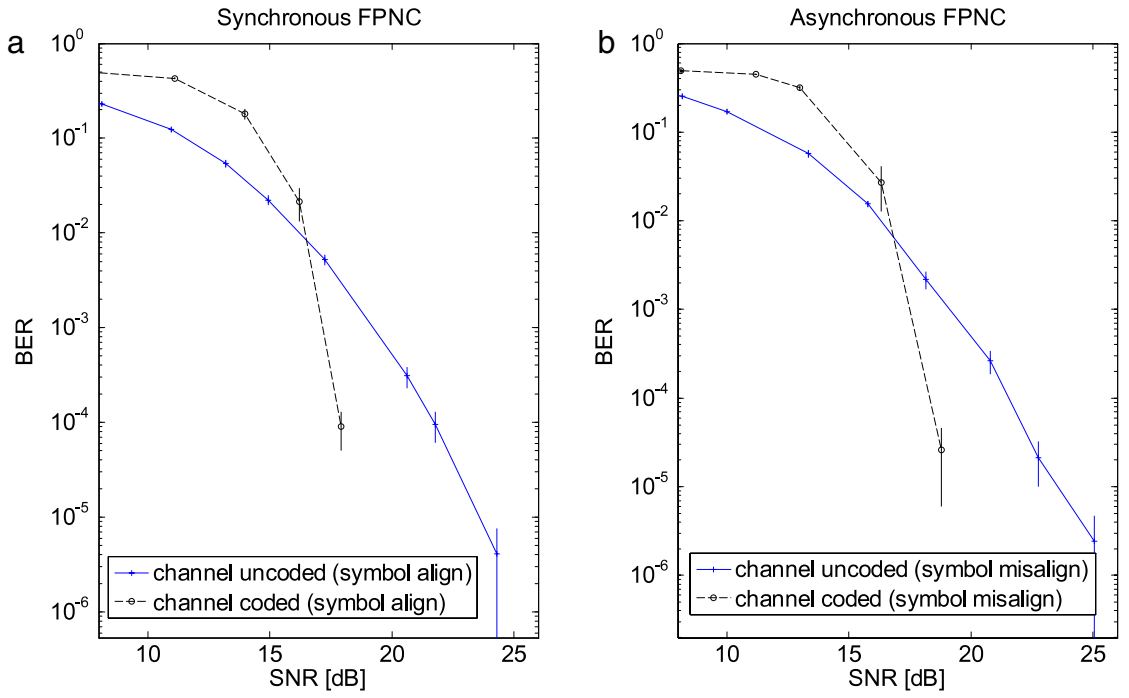


Fig. 13. BER of FPNC with and without sample synchronization. The 95% confidence intervals are marked in the figures. Note that the BER here is related to whether the XOR bit is decoded correctly, not whether the individual bits from the two end nodes are decoded correctly.

(2) $FER_{SNC} = P_f^{uplink,SNC}$: frame-error rate computed from the two uplink time slots in SNC.
(3) $FER_{P2P} = P_f^{P2P}$: frame-error rate of a point-to-point link.

Channel coded systems are considered in our implementation. All three systems use the convolutional channel code with 1/2 coding rate, as specified in the 802.11 a/g standard [9].

Note for (1) and (2), FER_{PNC} and FER_{SNC} refer to the error rate of the XOR of the two source frames. That is the error rate for the frame $\bar{S}_R = \bar{S}_A \oplus \bar{S}_B$. For FER_{SNC}, we gather the decoded $\bar{S}_A$ and $\bar{S}_B$ from the two uplink time slots, and then compute their XOR before checking whether there is an error in the XOR frame. For FER_{PNC}, the CNC scheme as described in Section 4.3 is used to decode $\bar{S}_R$ directly based on the simultaneously received signals. In Fig. 14(a), we plot FER_{PNC}, FER_{SNC}, and FER_{P2P} versus SNR obtained from our experiments. The FER measurements are all from channel-coded systems.

The throughputs per direction of the three TWRC schemes are computed as follows:

$$Th_{FPNC} = \frac{1}{2}(1 - P_f^{uplink,PNC})(1 - P_f^{P2P}),$$
$$Th_{SNC} = \frac{1}{3}(1 - P_f^{uplink,SNC})(1 - P_f^{P2P}),$$
$$Th_{TS} = \frac{1}{4}(1 - P_f^{P2P})^2. \quad (26)$$

In Fig. 14(b), we plot the throughputs (Th_{FPNC}, Th_{CNC}, and Th_{TS}) of FPNC, SNC and TS versus SNR based on the FER_{FPNC}, FER_{CNC}, and FER_{TS} in Fig. 14(a). With reference to Fig. 14(b), for the high SNR regime (above 19 dB), the throughput of PNC is approximately 99% higher than that of the TS scheme, and 49% higher than that of the SNC scheme. This is essentially the same as the ideal 100% and 50% throughput gains derived by slot counting in [1] (i.e., the error-free case), with the difference that we have channel coding here to ensure reliable communication. If we use the guideline that the common decodable 802.11 link usually works at an SNR regime that is higher than 20 dB [21,5], we can conclude that our FPNC implementation has very good performance in this regime. We note that for this regime, [5] mentions that ANC can achieve 70% and 30% throughput gains relative to TS and SNC. Hence, FPNC has better performance in this SNR regime by comparison.

We note from Fig. 14(b) that the performance of FPNC is not as good as that of SNC or TS at the lower SNR regime (say below 17 dB). This is most likely due to our specific implementation of FPNC in this paper rather than a fundamental limitation of FPNC in general. In particular, recall that we implement the CNC function in FPNC mapping (see Section 4.3) using the so-called XOR-CD approach. In XOR-CD, (i) we first perform XOR mapping for the channel-coded symbol pairs from the two end nodes; (ii) after that channel decoding is applied on the channel-coded XOR symbols to get the XOR of the source symbols. Step (i) loses information that could be useful for the decoding of the XOR of the source symbols, and may lead to inferior performance in the low SNR regime. This phenomenon is explained in [4,10], and a joint CNC scheme [4,22] for the PNC system can potentially achieve better performance than the XOR-CD

Please cite this article in press as: L. Lu, et al., Implementation of physical-layer network coding, Physical Communication (2012), doi:10.1016/j.phycom.2012.02.008

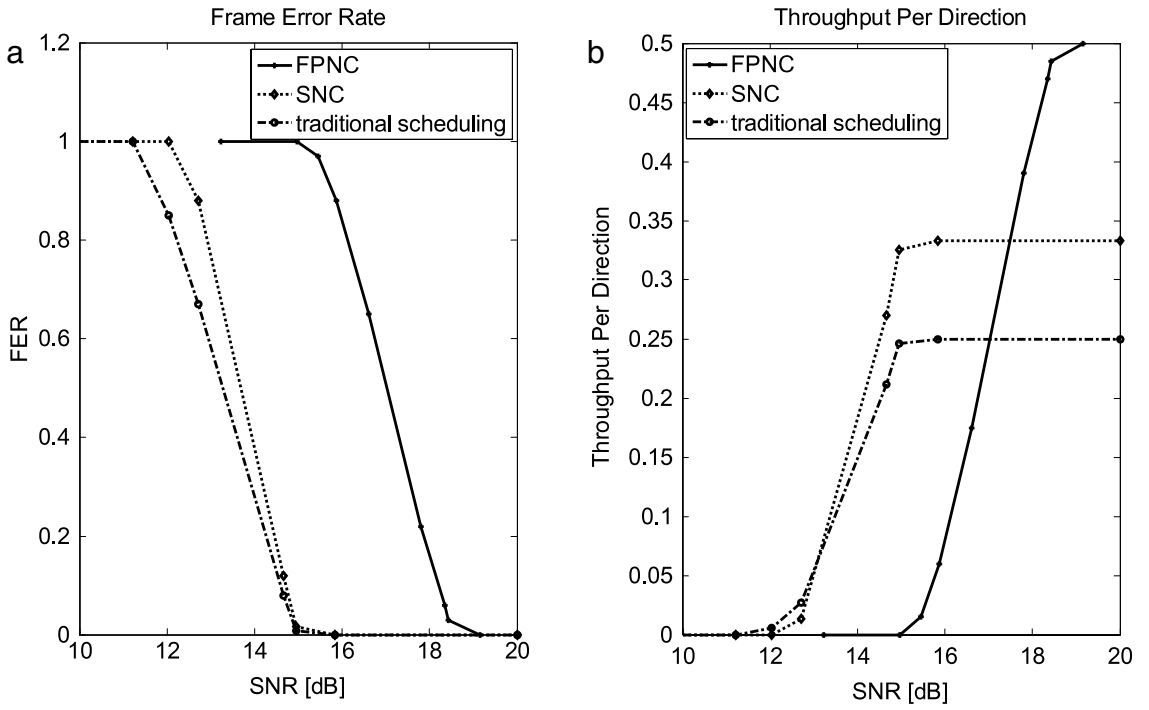


Fig. 14. Frame error rate and throughput comparison of FPNC with straightforward network coding and traditional secluding. (a) FER comparison of three approaches; (b) throughput comparison of three approaches.

scheme implemented in this paper, at the cost of higher implementation complexity.

6. Conclusion and future work

This paper presents the first implementation of a PNC system in which the relay performs the XOR mapping on the simultaneously received signals as originally envisioned in [1]. In particular, in our implementation, the XOR mapping is performed in the frequency domain of an OFDM PNC system. We refer to the OFDM PNC system as FPNC. The implementation of FPNC requires us to tackle a number of implementation challenges, including carrier frequency offset (CFO) compensation, channel estimation, and FPNC mapping.

A major advantage of FPNC compared with PNC in the time domain is that FPNC can deal with the different arrival times of the signals from the two end nodes in a natural way. We show by theoretical derivation that if the simultaneously received signals in FPNC have a maximum delay spread that is less than the length of the OFDM cyclic prefix (CP), then after the Discrete Fourier Transform, the frequency-domain signals on the different subcarriers are isolated from each other. That is, in the frequency domain, the signals are synchronous. Then, straightforward XOR mapping can be applied on the different subcarrier signals separately in a disjoint manner. To validate the advantage of FPNC, we present experimental results showing that time-domain symbol asynchrony does not cause performance degradation in FPNC.

To date, most work related to PNC focuses on its potential superior performance as derived from theory. In this paper, we evaluate the throughput gain of PNC relative to other two-way relay schemes. Our implementation indicates that PNC can have a throughput gain of 99% compared with traditional scheduling (TS), and a 49% throughput gain compared with straightforward network coding (SNC), in the high SNR regime (above 19 dB) in which practical technology such as Wi–Fi typically operates.

Going forward, there are many rooms for improvement in our FPNC implementation. In this paper, when faced with alternative design choices, we opt for implementation simplicity than performance superiority. For example, we choose to use a simple PNC mapping method called XOR-CD in this paper, which is simple to implement but has inferior performance compared with other known methods [4] in the low SNR regime. In addition, our implementation exercise reveals a number of problems with no good theoretical solutions yet, and further theoretical analysis is needed; in such cases, we use simple heuristics to tackle the problems. For example, CFO compensation for FPNC is an area that is not well understood yet, because we have to deal with CFOs of more than one transmitter relative to the receiver. In this paper, we simply compensate for the mean of the CFOs of the two end nodes. Better methods await further theoretical studies. Last but not least, we base our design on the 802.11 standard to a large extent with only moderate modifications. If we do not limit our design within the framework of 802.11, there could be other alternatives with potentially better performance.

Please cite this article in press as: L. Lu, et al., Implementation of physical-layer network coding, Physical Communication (2012), doi:10.1016/j.phycom.2012.02.008

References

[1] S. Zhang, S.C. Liew, P.P. Lam, Hot topic: physical layer network coding, in: Proc. ACM MOBICOM, 2006.
[2] R. Ahlswede, N. Cai, S.-Y.R. Li, R.W. Yeung, Network information flow, IEEE Trans. Inf. Theory 46 (2000) 1204–1216.
[3] S. Zhang, S.C. Liew, L. Lu, Physical layer network coding schemes over finite and infinite fields, in: Proc. IEEE Glob. Telecom. Conf., GLOBECOM, 2008.
[4] S.C. Liew, S. Zhang, L. Lu, Physical-layer network coding: tutorial, survey, and beyond, Physical Communication. (submitted for publication). Available at: http://arxiv.org/abs/1105.4261.
[5] S. Katti, S. Gollakota, D. Katabi, Embracing wireless interference: analog network coding, in: Proc. ACM SIGCOMM, 2007.
[6] F. Rossetto, M. Zorzi, On the design of practical asynchronous physical layer network coding, in: IEEE 10th Workshop on SPAWC, 2009.
[7] S. Zhang, S.C. Liew, P.P. Lam, On the synchronization of physical-layer network coding, in: Proc. IEEE Information Theory Workshop, 2006.
[8] L. Lu, S.C. Liew, S. Zhang, Optimal decoding algorithm for asynchronous physical-layer network coding, in: Proc. IEEE Int. Conf. on Comm., ICC, 2011.
[9] IEEE 802.11-2007. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, 2007. http://standards.ieee.org/getieee802/download/802.11-2007.pdf.
[10] S. Zhang, S.C. Liew, Channel coding and decoding in a relay system operated with physical-layer network coding, IEEE J. Sel. Areas Commun. 27 (5) (2009) 788–796.
[11] A. Oppenheim, R. Schafer, Discrete-Time Signal Processing, Prentice Hall, 2009.
[12] A. Goldsmith, Wireless Communications, Cambridge University Press, 2005.
[13] S. Glisic, Advanced Wireless Networks: 4G Technologies, second ed., John Wiley & Sons, 2007.
[14] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.
[15] B. Sklar, Digital Communications: Fundamentals and Applications, second ed., Prentice Hall PTR, 2003.
[16] G. fsf. gnu radio — gnu fsf project. http://standards.ieee.org/getieee802/download/802.11-2007.pdf.
[17] Ettus inc. universal software radio peripheral. http://www.ettus.com.
[18] P. Fuxjaeger, et al. Ftw ieee802.11 a/g/p ofdm frame encoder. https://www.cgran.org/wiki/ftw80211ofdmtx.
[19] P. Fuxjäger, A. Costantini, D. Valerio, P. Castiglione, G. Zacheo, T. Zemen, F. Ricciato, Optimal decoding algorithm for asynchronous physical-layer network coding, in: Proceedings of the IEEE 6th Karlsruhe Workshop on Software Radios, WSR.
[20] L. Lu, T. Wang, S.C. Liew, S. Zhang, Implementation of physical-layer network coding, in: Proc. IEEE Int. Conf. on Comm., ICC, 2012, pp. 1–7.
[21] J. geier. snr cutoff recommendations, 2005. http://www.wi-fiplanet.com/tutorials/article.php/3468771.
[22] L. Lu, S.C. Liew, Asynchronous physical-layer network coding, IEEE Trans. Wireless Commun. 11 (2) (2012) 819–831.

Lu Lu (S'10) received his B.E. Degree in Electronic Engineering and Information Science from the University of Science and Technology of China (USTC), Hefei, China, in 2007, and his Ph.D. Degree in Information Engineering from The Chinese University of Hong Kong (CUHK), Hong Kong, in 2012. He is now a postdoctoral fellow with the Institute of Network Coding and the Department of Information Engineering, CUHK. His current research interests include wireless communication in 802.11 networks, multi-user detection, collision resolution, physical-layer network coding and software-defined radios.

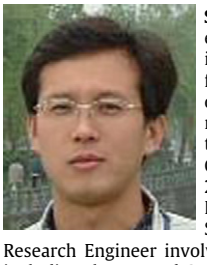
Taotao Wang received his B.S. Degree in electrical engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2008, and received his M.S. Degree in information and signal processing from Beijing University of Posts and Telecommunications, Beijing, China, in 2011. From Oct. 2010 to Jul. 2011, he was working as a research assistant with the Institute of Network Coding at the Chinese University of Hong Kong, Hong Kong. Now he is a Ph.D. student at the Department of Information Engineering of the Chinese University of Hong Kong. His main research topics are communication networks, wireless communication, signal processing, and information theory. His most recent research focus is on the application of network coding at the physical layer of communication systems. He is a recipient of the Hong Kong Ph.D. Fellowship 2011/12.

Soung Chang Liew (S'84-M'87-SM'93-F'12) received his S.B., S.M., E.E., and Ph.D. Degrees from the Massachusetts Institute of Technology. From 1984 to 1988, he was at the MIT Laboratory for Information and Decision Systems, where he investigated fiber-optic communications networks. From March 1988 to July 1993, he was at Bellcore (now Telcordia), New Jersey, where he engaged in broadband network research. He has been Professor at the Department of Information Engineering, the Chinese University of Hong.

Prof. Liew's current research interests include wireless networks, Internet protocols, multimedia communications, and packet switch design. Prof. Liew's research group won the best paper awards in IEEE MASS 2004 and IEEE WLN 2004. Separately, TCP Veno, a version of TCP to improve its performance over wireless networks proposed by Prof. Liew's research group, has been incorporated into a recent release of Linux OS. In addition, Prof. Liew initiated and built the first inter-university ATM network testbed in Hong Kong in 1993. More recently, Prof. Liew's research group has pioneered the concept of Physical-layer Network Coding (PNC).

Besides academic activities, Prof. Liew is also active in the industry. He cofounded two technology start-ups in internet software and has been serving as consultant to many companies and industrial organizations. He is currently consultant for the Hong Kong Applied Science and Technology Research Institute (ASTRI), providing technical advice as well as helping to formulate R&D directions and strategies in the areas of Wireless Internetworking, Applications, and Services.

Prof. Liew is the holder of eight US patents and a Fellow of IEEE, IET and HKIE. He currently serves as Editor for IEEE Transactions on Wireless Communications and Ad Hoc and Sensor Wireless Networks. He is the recipient of the first Vice-Chancellor Exemplary Teaching Award at the Chinese University of Hong Kong. Publications of Prof. Liew can be found in www.ie.cuhk.edu.hk/soung.



Shengli Zhang received his B. Eng. Degree in electronic engineering and the M. Eng. Degree in communication and information engineering from the University of Science and Technology of China (USTC), Hefei, China, in 2002 and 2005, respectively. He received the Ph.D. Degree in the Department of Information Engineering, the Chinese University of Hong Kong (CUHK), in 2008. From 2002 to 2005, he was with the Personal Communication Network and Spread Spectrum (PCN & SS) Laboratory, USTC, as a Research Engineer involved in several National 863 Research Projects including the Beyond-3 Generation of Mobile System in China (FUTURE Plan). From 2002 to 2005, he was also a Research Engineer of the UTStarcom Wireless Soft Research Center, Hefei, China, involved in the research and implementation of the WCDMA communication systems. From Oct. 2008 to May. 2009, he was a Research Associate in CUHK. Now, he is an Associate Professor with the Communication Engineering Department, Shenzhen University, China. His current research interests include wireless networks, wireless communication, physical layer network coding, and cooperative wireless networks.

Please cite this article in press as: L. Lu, et al., Implementation of physical-layer network coding, Physical Communication (2012), doi:10.1016/j.phycom.2012.02.008