

# Real-time Implementation of Physical-layer Network Coding

Lu Lu

Institute of Network Coding  
CUHK, Hong Kong  
lulu@ie.cuhk.edu.hk

Taotao Wang

Dept. of Inform. Engineering  
CUHK, Hong Kong  
wtt011@ie.cuhk.edu.hk

Lizhao You

Institute of Network Coding  
CUHK, Hong Kong  
lzyou@ie.cuhk.edu.hk

Minglong Zhang

Institute of Network Coding  
CUHK, Hong Kong  
mlzhang@ie.cuhk.edu.hk

Qing Yang

Dept. of Inform. Engineering  
CUHK, Hong Kong  
yq010@ie.cuhk.edu.hk

Shengli Zhang

Dept. of Comm. Engineering  
Shenzhen University, China  
zsl@szu.edu.cn

Soung Chang Liew

Dept. of Inform. Engineering  
CUHK, Hong Kong  
soung@ie.cuhk.edu.hk

## ABSTRACT

This paper presents the first real-time physical-layer network coding (PNC) prototype for the two-way relay wireless channel (TWRC). Theoretically, PNC could boost the throughput of TWRC by a factor of 2 compared with traditional scheduling (TS) in the high signal-to-noise (SNR) regime. Although there have been many theoretical studies on PNC performance, there have been few experimental and implementation efforts. We built the first prototype of PNC about a year ago. It was, however, an offline system in which an offline PNC decoder was used at the relay. For a real-time PNC system, there are many additional challenges, including the needs for tighter coordination of the transmissions by the two end nodes, fast real-time PNC decoding at the relay, and a PNC-compatible retransmission scheme (i.e., an ARQ protocol) to ensure reliability of packet delivery. In this paper, we describe a real-time PNC prototype, referred to as RPNC, that provides practical solutions to these challenges. Indoor environment experimental results show that RPNC boosts the throughput of TWRC by a factor of 2 compared with TS, as predicted theoretically. RPNC prototype provides an interface to the application layer, with which we demonstrate the exchange of two image data files between the two end nodes.

## Categories and Subject Descriptors

C.2.1 [Computer-Communication Networks]: Network Architecture and Design—*Wireless communication*

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

SRIF '13, August 12, 2013, Hong Kong, China.

Copyright 2013 ACM 978-1-4503-2181-5/13/08 ...\$15.00.

## General Terms

Algorithms, Design, Experimentation, Performance

## Keywords

Network coding, Physical-layer network coding, Soft demodulation, Realtime

## 1. INTRODUCTION

Since the introduction of the concept of PNC [1,2] in 2006, it has developed into a subfield of network coding with a wide following [3,4]. Figure 1 illustrates the concept of PNC when it is applied to a two-way relay channel (TWRC). In TWRC, two end nodes A and B exchange information via a relay R. The information exchange consists of two phases (time slots). In the first time slot, nodes A and B send signals simultaneously to relay R; in the second phase, relay R processes the superimposed signals of the simultaneous packets and maps them to a network-coded packet (XOR packet) for broadcast back to the end nodes. Each end node then extracts the packet from the other end node by subtracting its own packet from the network-coded packet. Theoretically, PNC could boost the throughput of TWRC by a factor of 2 compared with traditional scheduling (TS) in the high signal-to-noise (SNR) regime [1].

Although there have been many theoretical studies in the literature on PNC performance [3,4], there have been few experimental and implementation efforts. This paper demonstrates the first real-time implementation of PNC on a USRP software-defined radio (SDR) platform. We refer to this real-time PNC as RPNC. This is an advance from a previous non-real-time prototype reported by us in [5].

In the previous PNC prototype [5], signal processing and decoding were done offline. In addition, ARQ was not implemented. Furthermore, it was not a real functional communication system in that it lacked an application programming interface (API) and could not support real network applications. RPNC overcomes the challenges of fast real-time PNC signal processing and channel decoding; reduces the

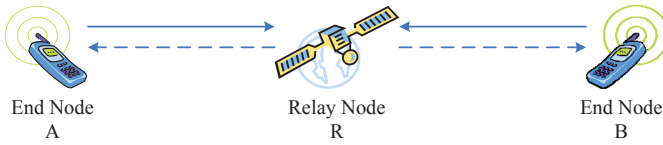


Figure 1: Two-way relay channel.

overhead due to the large latency between USRP and PC through a burst transmission mode; incorporates an PNC-compatible ARQ mechanism; and provides an API to support real network applications. With RPNC, we demonstrate the exchange of two image data files between two computers through a relay. In addition, we also demonstrate the doubling of throughput benchmarked against TS.

The remainder of this paper is organized as follows: Section 2 presents our solutions to various implementation challenges. Section 3 presents our experimental results. Section 4 concludes this paper.

## 2. IMPLEMENTATION CHALLENGES AND OUR SOLUTIONS

This section outlines the implementation challenges for a real-time PNC system. It gives an overview on how we address these challenges in the system design of RPNC.

### 2.1 Coordination and Synchronization of Simultaneous Transmissions

When the two end nodes transmit simultaneously to the relay in a PNC TWRC, their signals may arrive at the relay with symbol-time and carrier-frequency offsets (CFO) [6–8]. RPNC is an OFDM system. The symbol-time offset is not a major issue when OFDM is used. Specifically, RPNC, being an OFDM system, only requires the arrival times of the two packets from nodes A and B to be within the cyclic prefix (CP) of the OFDM signals [5, 9] (i.e., the delay difference of the two signals is within CP).

To ensure within-CP arrivals, our RPNC implementation makes use of beacons as in the 802.11 standard [10] to coordinate the local timers of the end nodes. The relay sends a beacon signal to nodes A and B to trigger them to transmit together. Since the transmissions of nodes A and B are triggered at the same time, the arrival times of the signals at the relay can be controlled to be within-CP.

Besides within-CP arrival times, the relative CFO between the two end nodes and relay node should be small enough so that the OFDM subcarriers are aligned with each other [5]. In RPNC, nodes A and B make use of CFO precoding to reduce the relative CFO. They use the beacon signal as a reference to synchronize their frequencies. Specifically, each end node first estimates the relative CFO between itself and the relay from the beacon. Then it compensates for the CFO by multiplying a compensating phase term to its signal before transmitting it to the relay. With this CFO precoding technique, we manage to reduce the relative CFO in our RPNC prototype rather substantially (the residual CFO is no more than 200Hz in our experiments).

### 2.2 Latency between USRP and PC

A well-known problem in GNU Radio is the large delay between the USRP and the host PC. Most of the signal processing tasks (e.g., demodulation, channel decoding) are

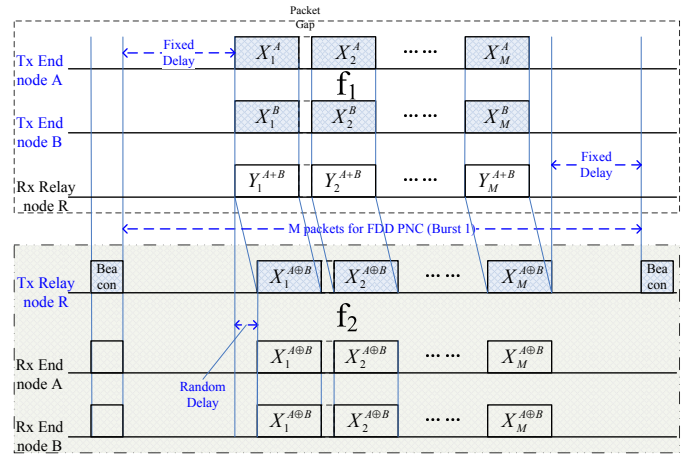


Figure 2: Burst mode transmission for RPNC. Note that the fixed delay indicated above may be large due to need to compensate for the large latency and large latency jitter between USRP and PC.

done at the PC. The time to hand over the samples back and forth between USRP and the PC is in general on the order of  $10^{-3}$ s. In addition, this delay is not fixed and has large jitters. The jitters are due to the unpredictable scheduling of CPU and other resources within the PC. For uplink transmission in RPNC, we need to align the arrival times of the two packets to within-CP, which is on the order of  $10^{-6}$ s (when the sample rate is 20MHz). Meanwhile, upon receiving the beacon, the two end nodes may take different amounts of time to process the beacon before their uplink transmission due to the aforementioned jitters. In particular, the two end nodes may experience different USRP-PC latencies. Thus, beacon triggering by itself may not guarantee within-CP alignment of the two signals.

To overcome the problem, we introduce an additional delay at the end nodes so that the total delay = beacon processing delay + additional delay. The additional delay is elastic so that it can compensate for the varying beacon processing time to achieve a fixed and constant total delay for both nodes. The two end nodes transmit simultaneously after this fixed total delay. Provided the target fixed total delay is large enough, the jitter can be smoothed out. As shown in [11], a  $10^{-2}$ s fixed delay is sufficient to synchronize different USRPs for simultaneous transmissions.

If each beacon triggers only one packet from each node in the uplink phase, the overhead of the large fixed delay will be rather significant. To reduce the overhead, in RPNC each beacon triggers a burst transmission consisting of multiple packets from each end nodes. To simplify design, we use frequency-division duplex (FDD) mode to isolate the uplink and downlink transmissions. In other words, data from the end nodes to the relay are carried on one frequency band, and the data from the relay to the end nodes are carried on another frequency band.

As shown in Figure 2, we leverage the FDD system setup to accelerate the feedback from the relay. That is, once the relay can successfully decode an uplink XOR packet, it will then broadcast the packet to the end nodes without switching the RF frontend from Rx mode to Tx mode.

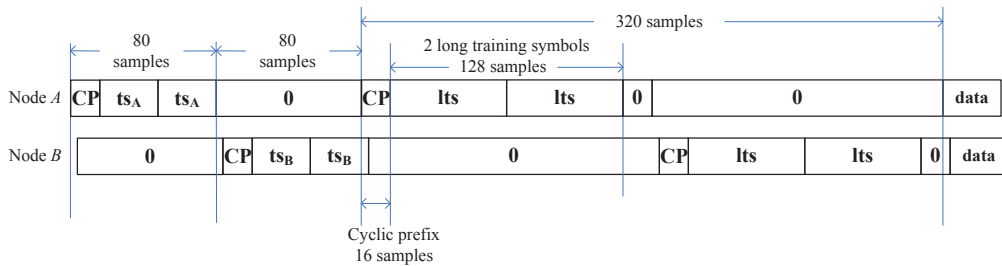


Figure 3: Frame format for RPNC.

### 2.3 PHY-layer Forward Error Control (FEC)

For reliable communication, channel codes need to be incorporated. RPNC adopts the convolution code as defined in the 802.11 standard. The goal of PNC channel decoding, performed at the relay, is to obtain a network-coded packet, which is an XOR of the source bits of packets from A and B [4]. For our RPNC prototype, the relay first maps the overlapped channel-coded symbols to their XOR on a symbol-by-symbol basis. After that it (i) channel-decodes the XOR channel-coded symbols to the XOR source symbols, and then (ii) re-channel-encodes the XOR source symbols to the XOR channel-coded symbols for forwarding to the two end nodes.

To increase the decoding success rate, we use a specially designed “soft” PNC channel decoding algorithm at the relay node. It adapts the standard single-user Viterbi algorithm for the purpose of decoding XOR source bits. Appendix A provides details of our soft PNC decoder design.

The relay also needs to do error checking to see if the aforementioned channel decoding has been successful or not. We modify the IEEE CRC-32 (cyclic redundancy check) function at the relay node to make its CRC function behave linearly with a property that  $CRC(A \oplus B) = CRC(A) \oplus CRC(B)$ , for the PNC uplink error checking.

### 2.4 MAC-Layer ARQ Error Control

For wireless communications (e.g., Wi-Fi), besides PHY-layer FEC, MAC-layer ARQ is needed to ensure that all packets are delivered. This is because the PHY-layer FEC does not guarantee 100% successful channel decoding. Thus, we need a MAC-layer ARQ that is compatible with RPNC.

There are two kinds of MAC-layer ARQ protocols for RPNC: end-to-end ARQ, where the relay does not store the uplink packets and does not participate in the ARQ mechanism; and link-by-link ARQ, where each of uplink and downlink operates an independent ARQ mechanism.

For end-to-end ARQ, there are two kinds of errors: the relay may fail to obtain the XOR packet in the uplink phase, or the relay may succeed in obtaining the XOR packet but the transmission of the XOR packet fails in the downlink phase. In either case, the end nodes A and B are responsible for retransmitting packets that fail to be delivered.

For link-by-link ARQ, the relay stores the XOR packets it successfully decodes, and if the downlink phase fails, the relay is responsible for retransmitting the XOR packet, not the end nodes. In other words, the uplink and downlink ARQs are decoupled so that the end nodes are responsible for retransmission of uplink packets, and the relay is responsible for retransmissions of downlink packets.

Our current RPNC prototype adopts the end-to-end ARQ because it is less complex than the link-by-link ARQ. In general, however, we expect the link-by-link ARQ to have better performance and our future prototype may incorporate that.

### 2.5 Frame Format for Packet Detection, Channel Estimation and CFO Estimation

In wireless communications, the PHY-layer frame is designed to have training symbols and pilots to facilitate packet detection, channel estimation and CFO estimation. Unlike single-user point-to-point communication, the relay in PNC receives simultaneously signals from the end nodes. The frame format has to be designed with that in mind. Figure 3 shows our frame formats for packets transmitted by nodes A and B. Note that their training symbols are orthogonal in time. This allows the relay to detect the boundaries of packets from A and B. In addition, the relay can also detect whether one user is transmitting or both users are transmitting during the uplink phase.

## 3. IMPLEMENTATION AND EVALUATION

This section presents our experimental setup and results.

### 3.1 Experimental Setup

Our system makes use of the USRP hardware [12] and the GNU Radio software with the UHD hardware driver [13]. We extended the RawOFDM single-user point-to-point OFDM transceiver software [14] for RPNC. The extensions include:

- a) Modifications of the single-user soft-input Viterbi algorithm (VA) software for PHY-layer channel decoding in RPNC.
- b) Modifications of the preamble and pilot designs. Different user nodes use orthogonal preambles and frequency domain pilots so as to enable multiuser signal presence detection, multiuser channel estimation, and multiuser CFO tracking and compensation at the relay.
- c) Phase precoding. Phase precoding at the user transmitters has been implemented to reduce the relative CFOs between the two users at the relay. The transmitters of the two users make use of the preambles in the poll frames from the relayAP to estimate CFOs for precoding purposes.
- d) CRC checking for PNC systems. The mathematics of CRC check for the single-user system with the 802.11 CRC design [10] cannot be directly applied to the PNC

system. We have modified the CRC for PNC error detection at the relay node.

For experimentation, we deployed three sets of USRP N210 with XCVR2450 boards [12] indoor to emulate a TWRC system, as shown in Figure 1. The relay node R uses 802.11 channel 1 (2.412GHz) to poll the two end nodes to transmit together at channel 11 (2.462GHz). BPSK modulation is used. Our experiments were carried out using 4 MHz bandwidth<sup>1</sup> at midnight to minimize the co-channel interference from nearby ISM band equipment.

Our RPNC prototype provides an API to the application layer. We have demonstrated the exchange of two image data files between two end nodes through RPNC<sup>2</sup>.

### 3.2 Performance Evaluation

We evaluate the PNC performance in an FDD TWRC. For benchmarking, we compare it with the traditional four-phase scheduling (TS) scheme in which each of the four directional links in the TWRC take turns to transmit. As with PNC, in our TS implementation, we also let a beacon from the relay trigger a burst of uplink transmissions, and we also use different frequency bands for uplink and downlink transmissions. The difference between PNC and TS is that PNC end nodes transmit uplink packets simultaneously, whereas TS end nodes transmit uplink packets at different times in an alternate manner. For the downlink phase, the relay in TS also transmits packets to end nodes A and B at different times.

We perform control experiments for different SNRs. The receive powers of packets from nodes A and B at the relay are adjusted to be balanced, with less than 1dB variation between the two powers. The relay transmits 100 beacons to trigger simultaneous transmissions. Each beacon triggers a burst of 1000 uplink transmissions. We count the number of downlink packets that are received successfully at the end nodes to compute the absolute throughput. For comparison, we use normalized throughput as the metric. The normalized throughput is defined as the absolute throughput divided by the number of packets received by each end node.

Figure 4 shows the performance of PNC and TS. As can be seen, PNC outperforms TS by a factor of 2 when  $\text{SNR} \geq 8\text{dB}$ . Note that at  $\text{SNR} \geq 10\text{dB}$ , PNC does not improve further. This is because we use BPSK modulation, and the packet reception rate is nearly 100% beyond 10dB in our experiment. To further boost throughput, higher-order QAM modulation could be considered in the future.

Note that the performance of TS is better than PNC at  $\text{SNR} < 8\text{dB}$ . The main reason is that we use the so-called XOR-CD PNC decoding scheme [4, 5] at the relay. This scheme is known to have low complexity but poor performance in the low SNR regime. Other more sophisticated PNC decoding schemes will yield better performance at the low SNR regime at the expense of higher complexity [4].

### 4. CONCLUSIONS

We have experimentally demonstrated a first real-time physical-layer network coding system, named RPNC, on the

<sup>1</sup>For smooth working of the real-time system, we did not use the full 20 MHz 802.11 bandwidth.

<sup>2</sup>The demo video can be found at [https://www.youtube.com/watch?v=HmRBm\\_IIBQQ](https://www.youtube.com/watch?v=HmRBm_IIBQQ).

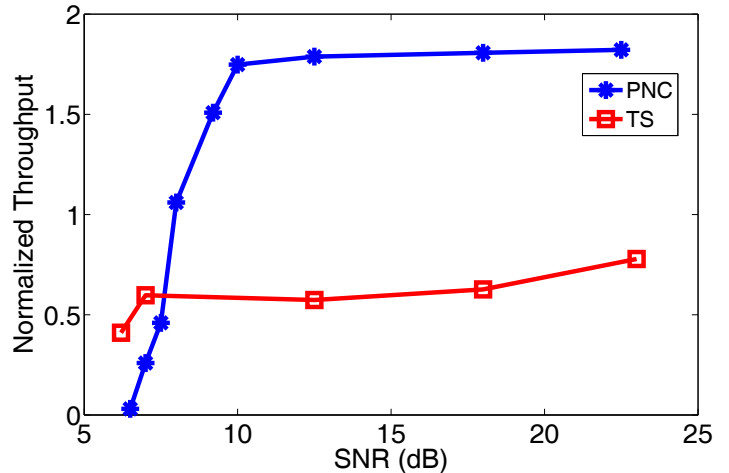


Figure 4: Normalized throughput of PNC and TS.

USRP software defined radio platform. Throughput gain of 100% relative to the traditional multi-hop scheduling has been achieved at  $\text{SNR} \geq 8\text{dB}$ . We believe our proposed concept has ample room for further advances. Interesting directions in the immediate future include:

- *Link-by-link ARQ* – In the current RPNC prototype, we have successfully implemented an end-to-end ARQ protocol. The ARQ performance could be further improved if we allow the relay node to participate in the ARQ process.
- *Higher-Order Modulation* – Our current system makes use of the BPSK modulation. When SNR is high, higher order QAM modulations could be considered. The theoretical extension of our PHY-layer decoders for higher-order QAM and experiment evaluation remains to be investigated.
- *Multi-way Relay Network* – Our current prototype assumes two-way relay channel. In a more general scenario, multiple end nodes could communicate through relays. How to identify PNC opportunities and perform low-complexity scheduling in the multi-way relay network is a promising direction for future work.

### 5. ACKNOWLEDGMENTS

This work is supported by AoE grant E-02/08 and the General Research Funds Project Number 414812, established under the University Grant Committee of the Hong Kong Special Administrative Region, China. This work is also supported by NSF of China (Project No. 61271277) and NSF of Shenzhen (Project No. JC201005250034A, JCYJ-20120613174214967).

### 6. REFERENCES

- [1] S. Zhang, S. C. Liew, and P. P. Lam. Hot topic: Physical layer network coding. In *ACM MOBICOM*, 2006.
- [2] P. Popovski and H. Yomo. The anti-packets can increase the achievable throughput of a wireless multi-hop network. In *Proc. IEEE Int. Conf. on Comm. (ICC)*, 2006.

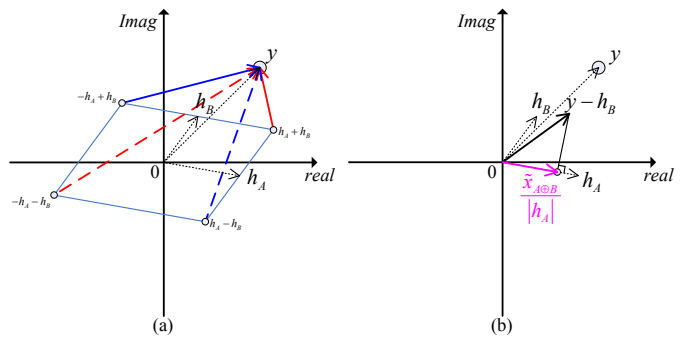
- [3] B. Nazer and M. Gastpar. Reliable physical layer network coding. *Proceedings of the IEEE*, 2011.
- [4] S.C. Liew, S. Zhang, and L. Lu. Physical-layer network coding: Tutorial, survey, and beyond. *Physical Communication*, 6(1):4–42, 2013.
- [5] L. Lu, T. Wang, S.C. Liew, and S. Zhang. Implementation of physical-layer network coding. *Physical Communication*, 6(1):74–87, 2013.
- [6] S. Zhang, S. C. Liew, and P. P. Lam. On the synchronization of physical-layer network coding. In *Proc. IEEE Information Theory Workshop*, Oct. 2006.
- [7] L. Lu, S. C. Liew, and S. Zhang. Optimal decoding algorithm for asynchronous physical-layer network coding. In *Proc. IEEE Int. Conf. on Comm. (ICC)*, Jun. 2011.
- [8] L. Lu and S. C. Liew. Asynchronous physical-layer network coding. *IEEE Trans. Wireless Commun.*, 11(2):819–831, Feb. 2012.
- [9] F. Rossetto and M. Zorzi. On the design of practical asynchronous physical layer network coding. In *IEEE 10th Workshop on SPAWC*, 2009.
- [10] IEEE 802.11-2009. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 5: Enhancements for higher throughput. <http://standards.ieee.org/getieee802/download/802.11n-2009.pdf>.
- [11] K. Wu, H. Tan, Y. Liu, J. Zhang, Q. Zhang, and L. Ni. Side channel: Bits over interference. In *ACM MOBICOM*, 2010.
- [12] Ettus Inc. Universal software radio peripheral. <http://www.ettus.com/>.
- [13] G. FSF. GNU Radio - Gnu FSF Project. <http://gnuradio.org/redmine/wiki/gnuradio>.
- [14] Szymon Jakubczak. Rawofdm: A raw I/Q-input OFDM and a fast QAM+FEC implementation for GNURadio. <http://people.csail.mit.edu/szym/rawofdm/README.html>.
- [15] J. Hagenauer and P. Hoeher. A viterbi algorithm with soft-decision outputs and its applications. In *IEEE Globecom*, 1989.
- [16] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

## APPENDIX

### A. SOFT VITERBI PNC CHANNEL DECODING

The PNC channel decoder aims to extract XORed packets from the two end nodes. A goal of ours is to simplify the decoder design so that real-time decoding is possible in our USRP prototype. Our RPNC prototype adopts the same convolutional code as that in 802.11.

Here, we adapt the standard Viterbi decoder for PNC decoding. For decoding of convolutional codes, the soft-input Viterbi decoding algorithm (VA) [15] has the advantage of being simple while having good performance. On the USRP platform, real-time decoding is possible with a VA decoder. However, the standard VA is meant for a single user system rather than PNC in which the goal is to extract the XOR of the packets. Here, we present a PNC decoding scheme in



**Figure 5: Illustration of the computation of the soft information on XOR bit: the solid and dashed lines of the same color represent the two Euclidean distances between the received sample and the two constellation points mapped to the same XOR value. (a) the solid blue line and the solid red line are selected to represent the Euclidean distances to the two different values of the XOR bit  $x_{A\oplus B}$ ; (b) projection of  $y - h_B$  onto  $h_A$  to get the soft information  $\tilde{x}_{A\oplus B}$ .**

which the standard VA decoder is used, with the difference being the soft information being fed to the the VA decoder as input. In particular, our scheme extracts soft information on the XOR bits.

In the following, the frequency-domain signals in OFDM are considered. Let  $X_A = (x_A[1], \dots, x_A[k], \dots)$  and  $X_B = (x_B[1], \dots, x_B[k], \dots)$  denote the modulated codewords (transmitted packets) of users A and B, and let  $\Pi(\cdot)$  denote the functional mapping corresponding to convolutional coding. These channel-coded symbols are obtained from the source symbols  $S_A = \{s_A[k]\}_{k=1,2,\dots}$  and  $S_B = \{s_B[k]\}_{k=1,2,\dots}$ , respectively. Since  $\Pi(\cdot)$  is linear, we have  $X_A \oplus X_B = \Pi(S_A) \oplus \Pi(S_B) = \Pi(S_A \oplus S_B)$ , where the XOR of two packets correspond to their bit-wise XOR. This means that we can first detect the XOR of individual bits  $\{x_A[k] \oplus x_B[k]\}_{k=1,2,\dots}$  and then feed the information on these XOR bits to the standard convolutional decoder for channel decoding [4]<sup>3</sup>. For our PNC VA, we feed the soft information on  $\{x_A[k] \oplus x_B[k]\}_{k=1,2,\dots}$  to the standard VA. For the subsequent discussion, for brevity, we will drop the index  $k$  in our notations. In addition, we will write  $x_A[k] \oplus x_B[k]$  simply as  $x_{A\oplus B}$ .

Our proposed  $\tilde{x}_{A\oplus B}$ , the soft bit for  $x_{A\oplus B}$ , follows the same reasoning as in the derivation of  $\tilde{x}$  for the standard single-user VA. In the standard VA,  $\tilde{x} = \log \frac{P_0}{P_1}$  is the log likelihood ratio, where  $P_0$  is the probability that the bit is 0 and  $P_1$  is the probability that the bit is 1.

When there are signals from both nodes A and B, the received signal is

$$y = h_A x_A + h_B x_B + n, \quad (1)$$

where  $h_A$  and  $h_B$  are channel gains of the received packets from nodes A and B at the relay, respectively, and  $n$  is Gaussian noise with variance  $\sigma^2$ . For  $\tilde{x}_{A\oplus B}$ , the log likeli-

<sup>3</sup>Note:  $x_A[k]$  and  $x_B[k]$  adopts value of 1 and  $-1$  rather than 0 and 1; therefore, we define  $x_A[k] \oplus x_B[k] = x_A[k]x_B[k]$ .

hood ratio is given by [1, 8]

$$\begin{aligned} \log \frac{P_0}{P_1} &= \log \frac{\Pr\{x_A=1, x_B=1|y\} + \Pr\{x_A=-1, x_B=-1|y\}}{\Pr\{x_A=-1, x_B=1|y\} + \Pr\{x_A=1, x_B=-1|y\}} \\ &= \log \left( \exp \left\{ -\frac{|y-h_A-h_B|^2}{2\sigma^2} \right\} + \exp \left\{ -\frac{|y+h_A+h_B|^2}{2\sigma^2} \right\} \right) \\ &\quad - \log \left( \exp \left\{ -\frac{|y+h_A-h_B|^2}{2\sigma^2} \right\} + \exp \left\{ -\frac{|y-h_A+h_B|^2}{2\sigma^2} \right\} \right). \end{aligned} \quad (2)$$

Unlike the single-user case, it is difficult to further simplify the log likelihood ratio expression above without making approximations. We adopt the log-max approximation,  $\log(\sum_i \exp(z_i)) \approx \max_i z_i$  [16]:

$$\begin{aligned} \frac{\sigma^2}{2} \log \frac{P_0}{P_1} &\approx \frac{1}{4} \max \left\{ -|y-h_A-h_B|^2, -|y+h_A+h_B|^2 \right\} \\ &\quad - \frac{1}{4} \max \left\{ -|y+h_A-h_B|^2, -|y-h_A+h_B|^2 \right\}. \end{aligned} \quad (3)$$

We can let  $\tilde{x}_{A\oplus B} = \frac{\sigma^2}{2} \log \frac{P_0}{P_1}$  given in (3) since the constant factor  $\frac{\sigma^2}{2}$  will not affect the shortest path found by VA.

Solution (3) can be further simplified. We illustrate this with an example. Fig. 5 shows the constellation map of a specific pair of channel gains,  $h_A$  and  $h_B$ . With the BPSK modulation,  $(x_A, x_B)$  takes on four possible values,  $(\pm 1, \pm 1)$ . In (3), either  $(+1, +1)$  or  $(-1, -1)$  will be chosen in the first  $\max(\cdot)$  function, and either  $(+1, -1)$  or  $(-1, +1)$  will be chosen in the second  $\max(\cdot)$  function. Which pair is chosen depends on  $y$ . In Fig. 5, we show a particular realization of  $y$  such that  $(+1, +1)$  and  $(-1, +1)$  are chosen (i.e., we have  $|y-h_A-h_B|^2 \leq |y+h_A+h_B|^2$  and  $|y+h_A-h_B|^2 \leq |y-h_A+h_B|^2$ ). The corresponding soft bit for  $x_{A\oplus B}$  is

$$\begin{aligned} \tilde{x}_{A\oplus B} &\approx -\frac{1}{4}|y-h_A-h_B|^2 + \frac{1}{4}|y+h_A-h_B|^2 \\ &= h_A \cdot (y-h_B). \end{aligned} \quad (4)$$

The intuition for (4) is as follows. The fact that  $(+1, +1)$  and  $(-1, +1)$  are chosen means we have already made a first decision that node B transmits 1. Thus, we should then decide the XOR value based on the probabilities of what node A transmits. The term  $y-h_B$  in (4) corresponds to subtracting from the received sample  $y$  the decision that node B transmits 1. The component that contains the signal from node A can therefore be defined as  $y_A = y-h_B$ . It can be shown that the dot product  $h_A \cdot y_A$  is simply the soft information for an equivalent single user case in which only node A transmits. For the standard single-user VA decoder, the soft information fed into the decoder is simply  $h \cdot y$ , where  $h$  is the single-user channel and  $y$  contains only signal from one user. Note that for the single user case, when BPSK is used, the component of  $y_A$  that is orthogonal to  $h_A$  contains noise only. The dot product in the complex plane extracts the component of  $y_A$  that contains signal from node A.

For other realizations of  $y$ ,  $\tilde{x}_{A\oplus B}$  can be found similarly. Specifically, there are four cases, as follows. If  $y$  is such that:

- $(+1, +1)$  and  $(-1, +1)$  are chosen, then  $\tilde{x}_{A\oplus B} \approx h_A \cdot (y-h_B)$ ;
- $(-1, -1)$  and  $(-1, +1)$  are chosen, then  $\tilde{x}_{A\oplus B} \approx -h_B \cdot (y+h_A)$ ;
- $(+1, +1)$  and  $(+1, -1)$  are chosen, then  $\tilde{x}_{A\oplus B} \approx h_B \cdot (y-h_A)$ ;
- $(-1, -1)$  and  $(+1, -1)$  are chosen, then  $\tilde{x}_{A\oplus B} \approx -h_A \cdot (y+h_B)$ .

Feeding the above soft information,  $\{\tilde{x}_{A\oplus B}[k]\}_{k=1,2,\dots}$ , to the standard VA allows us to decode for  $\{s_A[k] \oplus s_B[k]\}_{k=1,2,\dots}$ , which gives us the experimental results in Figure 4.