This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

Implementation and Evaluation of Cooperative Communication Schemes in Software-Defined Radio Testbed

Jin Zhang, Juncheng Jia, Qian Zhang and Eric M. K. Lo
Department of Computer Science and Engineering, Hong Kong University of Science and Technology
Email: {zjzj, jiajc, qianzh, ericlmk}@cse.ust.hk

***Abstract*—Cooperative communication is a promising technique for future wireless networks, which significantly improves link capacity and reliability by leveraging broadcast nature of wireless medium and exploiting cooperative diversity. However, most of existing works investigate its performance theoretically or by simulation. It has been widely accepted that simulations often fail to faithfully capture many real-world radio signal propagation effects, which can be overcome through developing physical wireless network testbeds.**

In this work, we build a cooperative testbed based on GNU Radio and Universal Software Radio Peripheral (USRP) platform, which is a promising open-source software-defined radio system. Both single-relay cooperation and multi-relay cooperation can be supported in our testbed. Some key techniques are provided to solve the main challenges during the testbed development: e.g., maximum ratio combine in single-relay transmission and synchronized transmission among multiple relays. Extensive experiments are carried out in the testbed to evaluate performance of various cooperative communication schemes. The results show that cooperative transmission achieves significant performance enhancement in terms of link reliability and end-to-end throughput.

I. Introduction

Cooperative communication is a promising technique to overcome fading and interference in wireless environment. It leverages the broadcast nature of wireless channel, and enables multiple wireless terminals to assist each other for high quality transmission. By combining signals through different paths from different users in receiver node, both spatial diversity and user diversity are fully exploited, which dramatically enhances system performance in terms of reliability and throughput. Fundamentally, cooperative communication can be viewed as a virtual multiple input multiple output (MIMO) system, which enables single-antenna devices to cooperate with each other to form a multiple antenna system. Through cooperation, it is then possible to obtain the cooperative diversity benefits of MIMO systems without the cost of having a physical antenna array at each terminal. Because of the above mentioned benefit, cooperative communication is a promising technique which can be widely used in next-generation wireless networks.

Since Cover and Gamal [1] proposed the concept of cooperative relay channel as a preliminary work in this direction, there are a lot of existing works proposing different cooperative communication schemes [2], [3] and investigating their performance in theory [4]–[6]. General performance bounds were also derived for various scenarios [7]. However, most of the works mentioned above analyze the network performance in theory or by simulation. Theoretical analysis is usually to provide guidance for real implementation, with many physical factors over simplified; meanwhile, it has been widely accepted that simulation based evaluations often fail to faithfully capture many real-world radio signal propagation effects. From the overall network point of view, how much performance enhancement cooperative communication can bring in real wireless environment is still an open question. To answer this question, a real cooperative communication testbed should be built to evaluate the overall network performance of different PHY and MAC layer schemes.

There are several existing platforms for testbed evaluation of cooperative communication. Some of them are based on the commodity wireless cards (e.g., 802.11 NIC) [8]. However, the key technologies of cooperative communication have special requirements of physical layer, which cannot be supported by these commodity wireless cards. Apart from commodity devices, some testbeds are based on digital signal processor (DSP) or field programmable gate array (FPGA) [9]. While such platforms can provide both necessary functions and high performance, the cost of development (specially the need for hardware programming) hinders its usage by research communities. Besides, the above testbeds don't provide basic functions to implement various MAC layer protocols. In contrary, software defined radio (SDR) based on general purpose processor is a promising way for development and configuration: most of the signal processing and MAC layer functions can be coded with much more efficient programming environment. USRP and GNU Radio [10], [11] is one of the most widely used SDR platforms these days. However, there are quite limited numbers of cooperative communication testbed based on such platform. The existing ones can only support quite limited functions [12], [13]. They can only do *selective cooperation*, which means the receiver receives the signal of the same series of packets from sender and relay one-by-one, and only select the signal with the best quality (e.g., the smallest bit error rate). More advanced schemes are not supported by the existing evaluation testbeds. Recent studies have demonstrated that *signal combination* by combining signals from the sender and relay at symbol level can fully exploit cooperative diversity. Even higher performance gain can be achieved if multiple relays can transmit the same packets simultaneously with strict time synchronization, which is called *synchronized multi-relay cooperation*. Using this technique multiple relays can forward the packet in the same slot instead of accessing channel in TDMA mode, so that signals from more independent paths can be leveraged in the receiver without extra time consuming, therefore, transmission efficiency can be further enhanced. It is pity that none of those advanced cooperative technologies have ever been evaluated in a real testbed.

We in this paper describe our recent work on testbed construction based on USRP and GNU Radio platform to overcome the above mentioned disadvantages. The key target is to design a reconfigurable and complete-function cooperative communication testbed to evaluate the real performance from both physical and MAC layers point of view. More specifically, we focus on single-relay cooperation with signal combination, and synchronized multi-relay cooperation. In our testbed, each node is composed of a USRP2 hardware and a general purpose host PC which runs GNU Radio with our designed cooperative communication library. To achieve strict time synchronization which is the base for synchronized multi-relay cooperation, we connect external global positioning system (GPS) signal to USRP2 board. To make such a system workable, we addressed several issues from both software and hardware sides, which will be the common issues for most of the USRP and GNU Radio based cooperative communication platforms.

1) Signal alignment and signal combination are the essential functions in the cooperative demodulation to fully exploit cooperative diversity. Signals from the sender and that from the relay arrive consecutively, the first copy of the signal needs to be stored, and aligned to and combined with the second copy of the signal using maximum ratio combine. We provide some function modules to achieve signal alignment and signal combination among signals from various paths.

2) Symbol level synchronization is a must for multi-relay cooperations with multiple relay's signal simultaneously reaches the receiver. Multiple relay nodes need to coordinately transmit packet so that the receiver can receive the same symbol from multiple paths at the same time, which is necessary for correct decoding. To achieve symbol level synchronization, timestamp methodology is introduced to achieve it by maintaining a hardware counter. Corresponding firmware and software modifications are implemented to achieve synchronization among multiple relays.

We believe the methodology we use in our testbed construction will be helpful for further research development of evaluating cooperative network performance on USRP and GNU Radio based testbed. We have the following contributions in this paper:

1) We design and implement a complete-function testbed framework based on USRP and GNU Radio to support cooperation communication, which includes a signal processing library for physical layer implementation of cooperative communication. Some key techniques such as signal alignment, signal combination and multi-relay synchronization are provided for future cooperative testbed construction.

2) We conducted extensive experiment to reveal the real system performance of cooperative communication with our platform. The result shows significant throughput gain of cooperative transmission compared with direct transmission, which is coherent with theoretical analysis.

The remaining of this paper is organized as follows: in Section II, several related works are reviewed. In Section III, hardware and software architecture about GNU Radio and USRP testbed is introduced. In Section IV and V, the main challenge and implementation method to do single-relay cooperation and multi-relay cooperation is described respectively. Experiment result and analysis is given in Section VI, and finally in Section VII is the conclusion of the whole paper and future work.

II. Related Works

There are several existing works on building testbed to evaluate cooperative diversity in real wireless systems. We first review some cooperative testbeds in terms of their designs and implemented functions. Then USRP and GNU Radio based works are summarized and compared with our testbed.

Some cooperative communication testbeds are based on the off-the-shelf wireless cards. For example, in [8] CoopMAC protocol is implemented in an open source driver for 802.11 devices, and a comprehensive set of experiments are conducted in a testbed consisting of up to 10 stations. Similarly, in [12] simple commodity hardware were used to create cooperative relays, where cooperation were limited to simple selection of packets from sender or relay, without signal combination. The benefits of diversity were observed qualitatively by noting that the display at the destination had noticeably fewer errors in the text when cooperation was employed. This is mainly because commodity hardware can not be easily modified at MAC layer and almost impossible at physical layer, where cooperative communication technology is located.

On the other hand, some testbeds are based on digital signal processor (DSP) or field programmable gate array (FPGA). In [9], a DSP based testbed is built with laptop PC for control. Well-known cooperating relaying schemes, amplify-and-forward (AF), detect-and-forward (DF), cooperative maximum ratio combining (CMRC), and distributed space-time coding (DSTC), are implemented with such a testbed. While such platforms can provide both necessary functions and high performance, programmability is usually poor for such testbeds: developers need to be familiar with low-level hardware programming with specific developing environment.

Recently, there are some emerging testbed development platforms, which are suitable for cooperative communication research and development. One of them is WARP [14] by Rice University, which provides flexibility for both lower layer hardware and upper layer software programming. In [13], an amplify-and-forward network was constructed based on an OFDM physical layer and distributed Alamouti transmit diversity scheme using WARP. However, it is relatively hard to extend single relay to synchronized multi-relay cooperation in WARP, as no external clock interface are provided in hardware. Another concern for WARP platform is its price.

Besides, the platform of USRP and GNU Radio is becoming hot. By implementing all signal processing in software, it

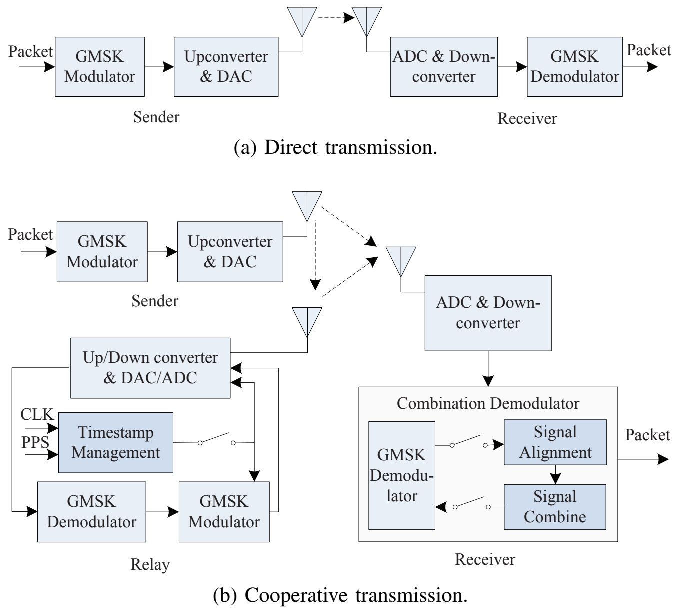


Fig. 1: Function blocks of software-defined radio nodes in direct and cooperative transmission.

can provide quick development and easy reconfigurability in radios thus is suitable to build cooperative communication testbed on it. The open nature of the GNU Radio project made it possible to leverage the work already done by others and provided an opportunity to add functionality of which others could easily take advantage. Due to its open source, flexibility and popularity, a lot of significant novel schemes have been proposed and verified on GNU Radio based testbed, including Analog Network Coding [15], ZigZag [16] and MIXIT [17]. Besides being used to verify new proposed cross-layer novel techniques, GNU Radio is also used to evaluate the performance of multi-antenna system, Kim et. al. [18] build Hydra prototype on GNU Radio testbed and evaluates the performance of MIMO system and propose rate adaptation protocol based on the experimental results. In [19], a GNU Radio based testbed is used to evaluate the performance of simple DF and selective DF, without further design and development of signal combining. Until now, there is no cooperative communication module provided in GNU Radio based testbed, and several key technologies have not been realized on this platform because of implementation issues such as time synchronization. As a result, little experiment evaluation is done on such a testbed. In comparison, our testbed can achieve maximum ratio combine in symbol level when doing single relay cooperative transmission. Synchronized multi-relay cooperation is also supported in our testbed, which is not supported in existing testbeds.

III. Hardware and Software Architecture

Our testbed supports both single-relay cooperation and multi-relay cooperation. To demonstrate single-relay cooperation, the testbed consists three nodes: one sender, one relay and one receiver. To demonstrate multi-relay cooperation, at least four nodes are needed: one sender, two relays and one receiver. Each node is a software-defined radio nodes, which consists an RF-frontend implemented in Universal Software Radio Peripheral 2 (USRP2) [11] board, and a signal processing module implemented in GNU Radio running in general purpose computer.

We use USRP RFX2400 daughterboard as RF-frontend to provide filtering of the RF signal and conversion from RF to IF and vice-versa. RFX2400 is a transceiver capable of operating in the 2.3 GHz to 2.9 GHz range with a peak output power of 50 mW. The output signal is then connected to the USRP2 motherboard, where it is sampled by an ADC and then be converted to baseband by a digital downconverter (DDC) implemented in the onboard FPGA. The transmission path is similar, but consists of digital upconverters (DUC) and a DAC. The baseband digital signal out of USRP2 motherboard is sent via Gbit Ethernet interface to the host computer running GNU Radio to do physical layer signal processing.

Existing GNU Radio provides GMSK module to support two-node direct transmission using GMSK modulation. As shown in Figure 1.(a), in sender, the packet from upper layer is processed by the GMSK modulator into the modulated baseband complex signal, which is then processed by USRP board before sending it to RF antenna. In the receiver side, the RF signal is transformed to digital baseband signal in hardware and then processed by GMSK demodulator until they are decoded into packets.

To support single-relay cooperation, the forwarding function is added in relay node, as shown in Figure 1.(b), which combines GMSK demodulator originally in receiver and GMSK modulator in sender together. In receiver, two key components, signal alignment and signal combination module, are added to collaborate with existing demodulator to achieve signal combination at symbol level so as to fully exploit cooperative diversity.

For synchronized multi-relay cooperation, except for forwarding function, timestamp control module is added to synchronize multiple relay's forwarding packets. The key target of this module is to enable the hardware to send packets at specified time. We modify the firmware in FPGA in USRP2 motherboard to maintain a hardware mainclock-based counter, which output timestamp information. Timestamp-aware receiving and sending is also enabled. We leverage two external signals 10MHz clock (CLK) and pulse per second (PPS) to calibrate the counter at different relays. Software signal processing blocks in GNU Radio is also modified to add timestamp control functionality. The receiver for synchronized multi-relay cooperation is the same as directly transmission, as the signal is already combined in wireless channel. In both relay and receiver, there is a switch to enable the node to switch between single-relay mode and multi-relay mode.

IV. Single-Relay Cooperation with Signal Combination

In this section, we introduce the main challenge and implementation method of conducting single-relay cooperation with signal combination.

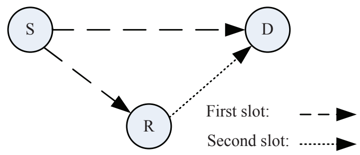


Fig. 2: Three-node cooperative transmission.

Usually for such cooperation, the whole transmission is divided into two slots as shown in Figure 2. The sender sends the packet in first time slot and the relay forward it in the second slot, after receiving two signals from different paths, the receiver combines them together to decode the packet. The main challenge is how to combine the two copies of the signals transmitted from different paths in the receiver side.

Various physical layer modulation schemes can be used by signal combination. We in this paper focus on the GMSK modulation. The GMSK signal is demodulated by differential phase detection (DPD) in our testbed. DPD is a noncoherent detection which can avoid channel estimation and reduce the demodulation complexity. It calculates the product of one symbol with the conjunction of its previous symbol,

$$Z = X(kT + T)X^*(kT) = A_Z e^{j\theta_Z},$$

where $X(kT)$ is the transmitted signal corresponding to the k^{th} symbol and T is the duration of one symbol. The bit is decoded according to the phase of the calculated product,

$$\hat{\alpha_k} = \begin{cases} 1 & if \quad \theta_Z \geq 0 \\ 0 & if \quad \theta_Z < 0. \end{cases}$$

Assuming there are two copies of the signal sent from the sender and relay respectively, which are denoted as X_1 and X_2, then to leverage maximum ratio combine, sum of the product of signals from both paths is calculated,

$$D = X_1(kT+T)X_1^*(kT)+X_2(kT+T)X_2^*(kT) = A_D e^{j\theta_D}.$$

The original bit is decoded according to the phase of the calculated summation,

$$\hat{\alpha}_k^* = \begin{cases} 1 & if \quad \theta_D \geq 0 \\ 0 & if \quad \theta_D < 0. \end{cases}$$

A. Challenges

Existing GMSK demodulation procedure for direct transmission is shown in Figure 3. The baseband digital signal from USRP board goes through the following GNU Radio signal processing blocks in sequence until it is decoded into packets: phase demodulation module, which calculates the phase of each symbol from input signal stream in complex format and output a stream of real number; clock recovery module, which recovers clock to achieve symbol synchronization based on phase information, and outputs one real number presenting the phase of the symbol per symbol time; decision module, which rounds the phase information to bit 0 or 1; access code correlator, which uses a known packet access code to do correlation with the input 0/1 bit sequence to find the starting point of each packet, and finally frame sink module, which resembles the 0/1 bit sequence to packets and do CRC checksum to check whether there is bit error in the packet.

The key challenges in cooperative decoding is signal alignment and signal combination:

1) To exploit spatial and user diversity, two signals transmitted from different paths are combined before decoding. To guarantee correct decoding, the starting point of the two transmitted packets need to be identified and the signal from one path should be combined with the other path's signal corresponding to the same symbol, which is called *signal alignment*. It can only be done after access code correlator, as only after the signals are correlated with the known access code, the start of the packet can be identified.

2) There is no signal combination functions in existing GNU Radio library. Besides, traditional signal combination is merely a physical layer approach. Actually, for the case here, if we correctly receive the packet from either sender or relay, there is no need for further signal combination, which is a combination of both selective cooperation approach and signal combination approach. Again, there is no existing functions now.

We will explain our method in detail to solve the above two problems in the following two subsections.

B. Signal Companion for Signal Alignment

To guarantee two signals corresponding to the same symbol to be combined, we propose signal companion method to achieve signal alignment. In first slot, while the receiver is trying to decode the signals from the sender, it should meanwhile keep an copy of the original signal steam in complex format together with the decoded signal flowing through each modules as shown in Figure 4, which is called *signal companion*. In second slot, the original signal from the relay should also be kept during the signal processing procedure. We align the two flows of original signals basing on the result of the access code correlation block. The start of a packet can be found by correlating the signal with a known access code, when the start point of both signals from sender and relay are found, the signal alignment is achieved and the two signals can be combined.

To keep record of the original signal, every module in front of the combining and decision module is added an additional input port and output port to receive and transmit the raw signal stream in complex format respectively, as shown in Figure 4. Such design makes less modification on the original flow graph.

Using access code correlator to achieve signal alignment makes one underline assumption that the access code need to be decoded correctly. As it is only 8-bit long, its error probability is quite low compared with the payload, which is much longer. In our experiment, we found that this factor can be ignored at medium to high SNR, or transmission power equivalently. One solution to correct access code error in low SNR area is to transmit access code in low modulation rate while using high modulation rate to transmit data, which will be investigated in future works.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

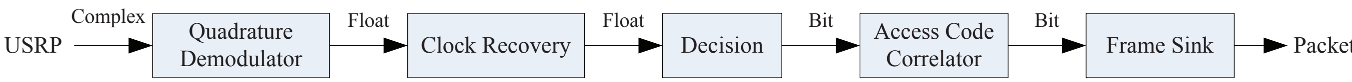


Fig. 3: Receiving Signal processing flow graph for direct transmission.

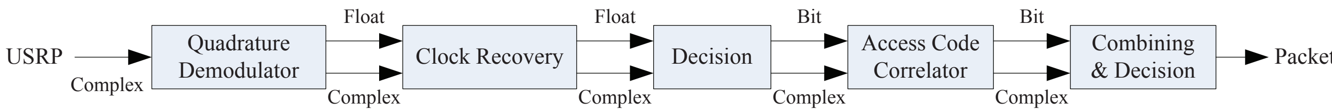


Fig. 4: Receiving signal processing flow graph for cooperative transmission.

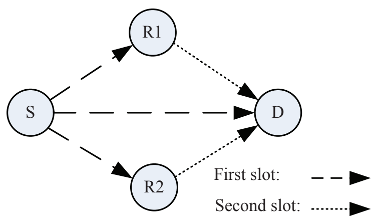


Fig. 5: 4-node multi-relay cooperative transmission.

C. Selective Maximum Ratio Combine for Signal Combination

Signal combination is the most important issue in cooperative transmission, which can fully exploit user and spatial diversity. In our implementation, signal combination and decision are conducted in the last block of the flow graph as shown in Figure 4. For a packet transmitted from the sender, if it is successfully decoded (pass the CRC check), the packet will be passed over to upper layer directly; otherwise, the input complex signal is stored in temporary and waits for relay's signal. For a packet from the relay, if it is successfully decoded, it will be passed over to upper layer; otherwise, it will be combined with previous stored signal from sender using maximum ratio combine according to the maximum ratio combine theory described in Section IV. The experimental result shows that the reliability is enhanced due to the user and spatial diversity exploited by maximum ratio combine.

V. Multi-Relay Cooperation

Besides three-node single-relay cooperation, multi-relay synchronized cooperation is also supported in our testbed to fully exploit user and path diversity from multiple users. The basic scenario is shown in Figure 5, there are four wireless nodes in the experiment: one sender, one receiver and two relays. The sender sends a packet in the first slot, the two relay nodes decode and forward it in the second slot. The receiver decodes the packet based on the signals it receives from both relays in the second time slot. We will introduce the key challenges and our solution in the following subsections.

A. Key Challenge: Time Synchronization

The most significant challenge in implementing the synchronized multi-relay cooperation is to achieve synchronization among multiple relays. As multiple relays transmit signals in the same channel at the same time, the receiver can not separate them individually. Instead, it receives a superposition of the two signals after path fading. If and only if the signals from multiple paths corresponding to a certain symbol arrives to the antenna of receiver exactly at the same time, the receiver can decode the symbol; otherwise, the two transmissions will interfere with each other, and neither of them can be decoded.

The above mentioned synchronization in the receiver can be achieved by synchronizing transmission from multiple relays for indoor environment. The distance between two nodes in indoor scenarios is only several meters. Compared with the symbol duration, the delay differentiation from different relay nodes to the receiver can be ignored. In that case, to achieve synchronization is to control multiple distributed relay nodes to sent a packet in the same time. We achieve this by introducing a novel timestamp methodology. This methodology can also be used in our-door environment, if we compensate the delay difference between different relays.

Currently, USRP sends the signals immediately to the RF-end once it receives signal from GNU Radio via Ethernet interface. Therefore existing GNU Radio testbed can only control the packet transmission time by delaying in software. However, as the random queuing delay generated in Ethernet in different relay nodes is usually in unit of microsecond, it's hard to achieve synchronized transmission in RF-end at symbol level by current software delay control. Therefore, our timestamp methodology achieves synchronization by controlling hardware configurations.

B. Timestamp Methodology

The key idea of timestamp methodology for hardware synchronization is as follows. A counter is maintained in each relay's USRP motherboard, whose value is added by one every main clock cycle. This clock is the FPGA mainclock in motherboard instead of the clock in general purpose processor, so we call it hardware synchronization. For each received sample in baseband, we record the corresponding counter value when it is sampled, which is called timestamp. The timestamp is passed through every demodulation module in GNU Radio until a packet is decoded. After receiving a packet, multiple relays add a same predefined delay to the received timestamp, and send the packet when the counter equals to the new timestamp. As the packet is received at the same time by multiple relays, after a constant delay, the packet is guaranteed to be sent at the same time, synchronization is then achieved.

To implement the above mentioned idea, we modify the verilog code in FPGA in USRP board to maintain a counter, to add timestamp information while receiving and to transmit signal at desired timestamp. We also modify the signal processing modules in GNU Radio to decode the timestamp information in receiving chain and generate required timestamp information in sending chain.

1) Firmware Modification in RF-frontend: Each relay maintains a counter based on their main clock. To guarantee time accuracy, frequency shift of the main clock is not allowed. Therefore, we provide each relay node an external clock to let the main clock locked to it. In our testbed, the main clock of USRP2 motherboard is 100MHz, multiple relay nodes are locked to an identical 10MHz clock. We also provide an external pulse per second (PPS) signal, which generates one pulse per second, to reset the counters of multiple relays at the same time. Both 10MHz clock and PPS signal are provided by OSA5200B GPS clock [20] in our experiment. The GPS clock generator uses GPS signal received from the BPS antenna as the input and provides four 10MHz clock and four PPS signals as output.

We modifies the verilog code in firmware to lock USRP2 board on the 10MHz clock and maintain a counter in the USRP2 motherboard. When transferring a block of signal from USRP to general purpose computer through ethernet, pack the timestamp information in the block to indicate the counter value when doing sampling. Provide a function called sync_to_pps to let the counter set to be zero when there is a rising edge in PPS signal. This function will be called by software when doing setup.

2) Software Modification in Signal Processing Blocks: To enable the timestamp methodology described in previous subsection, both transmitting and receiving signal processing flow graphs should be modified.

Figure 6 shows the demodulation flow graph in the receiving side. Compared with traditional demodulation flow graph, one additional signal indicating the timestamp is added in every module. The u_source has two output, one is the sampled signal, another is the timestamp corresponding to the sample. Be noted that USRP2 only provides one timestamp per block of samples, u_source needs to add a value of decimal number to each sample to indicate the real timestamp of each sample instead of each block. Then the timestamp is passed through all the following signal processing blocks until the packet is decoded. Some of the blocks are synchronized block (which means that the input number of items equals to the output number of items), for example, quadrature demodulator block, decision block and access code correlator block. In that case, outgoing timestamp is simply a copy of the ingoing timestamp information. The other blocks has different output number of items with input number of items, e.g., clock recovery. In that case, the outgoing timestamp should be interpolated or decimated to keep coherent with the output signal.

Figure 7 shows the modulation flow graph in the sending side. Similar with the receiving side, two additional signals are added to each block. One is timestamp, which indicates the desired timestamp the packet should be sent to the wireless channel. Another is the control bit, which indicates the start or the end of a burst. When the main program call the send_pkt function to send a packet, it need to set a timestamp together with the packet, and meanwhile, set start_of_burst to be 1. Then the timestamp will go through all the sending processing blocks until it arrives USRP2 board. The blocks in sending side are as follows in sequence: packet from upper layers are first passed through bits to symbols block to be converted to symbols, then, the symbols are shaped by a gaussian filter which is essential for GMSK modulation, the shaped signal in float format is then modulated by the frequency modulator block. The modulated complex signal, through amplifier, can be sent to USRP2 board via Ethernet interface. Meanwhile timestamp and control information is also sent to the board. The USRP2, after receiving complex signals and timestamps, will waits until the counter value equals to the timestamp and start_of_burst equals to 1, and then sends the complex signal to the RF-frontend.

The delay between receiving timestamp and the corresponding transmitting timestamp is set to be 1000000, which equals to 0.01 second for a 100MHz main clock. It is because in our experiment, we found that the signal processing time consumed by decoding and then generate a packet will be no larger than 0.01s. Therefore, setting the timestamp delay to be 1000000 can guarantee that when the regenerated packet is sent to USRP2, the counter will not run beyond the timestamp we prescribed.

VI. Experimental Results

In this section, experiments are conducted to evaluate the physical layer BER performance and MAC layer throughput and delay performance of various cooperative schemes.

All the experiments are conducted in in-door environment. USRP2 motherboard and RFX2400 daughterboard is used as the RF-frontend, which works on 2.4GHz. GNU Radio is implemented in dual-core 3.2G general purpose processor under Linux operating system. The Ubuntu version is 9.04 and GNU Radio version is 10955. GMSK modulation and demodulation is used for all experiment. The other system parameters used in the experiment for signal transmitting and receiving is listed in Table I. A power meter is used to measure transmission power, which is connectted to the coaxial RF connector of the sender without antenna. By adjusting the parameter tx-amplitude, different transmission powers can be achieved and a corresponding table can be created in preparation for further experiments.

A. Physical layer Performance Evaluation

For physical layer performance evaluation, we mainly concern the relationship between BER and transmission power under certain transmission rate for a single link. Experiment is done seperately for single relay cooperation and multi-relay cooperation.

1) Single-Relay Cooperation: Performance evaluation in three-node single-relay cooperation is conducted under the following two scenarios. Scenario 1 (shown in Figure 8.(a)): the sender, relay and receiver are located in the corners of an equilateral triangle. The distance between every two nodes

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

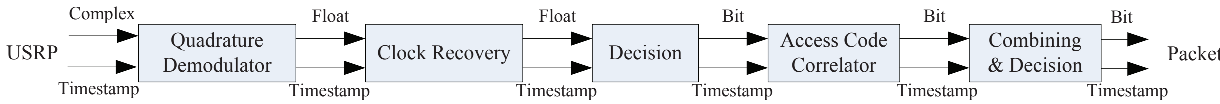


Fig. 6: Packet-receiving flow graph of relay node in multi-relay cooperation.

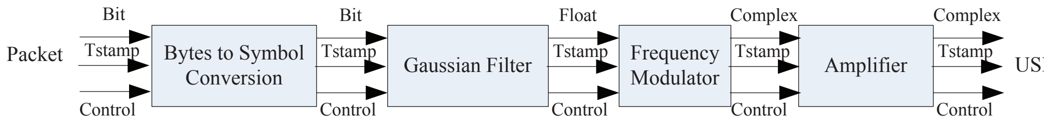


Fig. 7: Packet-transmitting flow graph of relay node in multi-relay cooperation.

(a) Single-relay scenario 1. (b) Single-relay scenario 2. (c) Multi-relay transmission.

Fig. 8: Experimental scenarios.

TABLE I: System Parameters.

Modulation	GMSK
Bit Rate	200,400,600, 800 Kbps
Hardware interpolation	250,125,83,62
Hardware decimation	250,125,83,62
Sample per symbol	2
Center frequency	2.4GHz
Packet size	1500 byte
Access code size	8 byte
Roll off	0.35
μ in time recovery	0.5

is 1.5 meter. A thick metal board is put between the sender and receiver to function as an obstacle to reduce the link quality. Scenario 2 (shown in Figure 8.(b)): the sender, relay and receiver are located in a straight line. Relay node is in the middle of the sender and receiver. The distances from relay to both sender and receiver are about 1.5 meter. All the nodes are fixed during the whole experiment.

We investigate the relationship between BER and transmission power under different transmission rate. The rate is set to be four levels: 200Kbps, 400Kbps, 600Kbps, 800Kbps. For a certain transmission rate, transmission power of the sender and relay is changed simultanuously so that the BER is in the range from 10^{-7} to 0.5. For each power level, 10000 packets are sent. For each packet, the sender transmits it in the first timeslot, then, the relay decode and forward it, the receiver does three kinds of decoding: decoding from the sender's signal, from the receiver's signal and from the combined signal. The destination calculates the bit error rate (BER) for three kinds of decoding individually.

Figure 9 shows the BER versus transmission power in scenario 1 for three transmission modes (D denotes direct transmission, R denotes relay transmission, C denotes combined transmission) under various rates. The channel from sender to receiver is extremely bad due to the obstacle in the channel, therefore, direct transmission has rather high BER. The BER of cooperative transmission is almost the same as (only a little bit lower than) that of relay transmission. It is because the direct transmission from the sender is too terrible compared with the relay transmission. Although path diversity is exploited, as the contribution of the sender's path is so trivial that can be neglected, the performance gain through combination is limited. Compare the performance of 400Kbps with that of 800Kbps for the same mode, it is obvious that high rate transmission requires larger transmission power to guarantee the same BER performance, which is in accordance with modulation theory.

Figure 10 shows the BER performance in scenario 2. It is obvious that performance of cooperative transmission achieve significant performance gain over both direct transmission and relay transmission, which is coherent with the theoretical analysis of maximum ratio combine. From this figure, we can draw another observation that cooperative transmission can save transmission power of the whole network. When the transmission rate equals to 800Kbps, using direct transmission, when the sender transmits at -17dbm, the BER at the receiver is 7.2×10^{-3}. Using cooperative transmission, when both sender and relay transmit at -20dbm, which keeps the sum of the power of both sender and relay equals to that of the sender in the direct transmission, the BER at the receiver is about 4.1×10^{-3}, which is lower than direct transmission, which

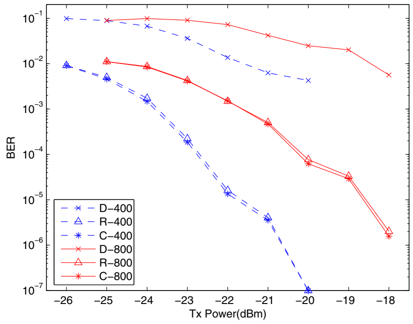


Fig. 9: Performance comparison of three-node cooperation for scenario 1.

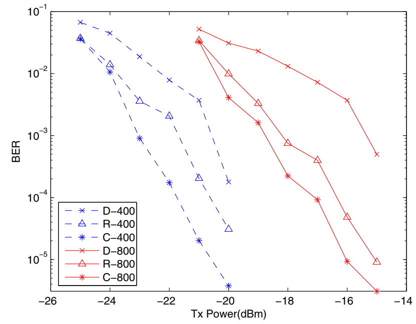


Fig. 10: Performance comparison of three-node cooperation for scenario 2.

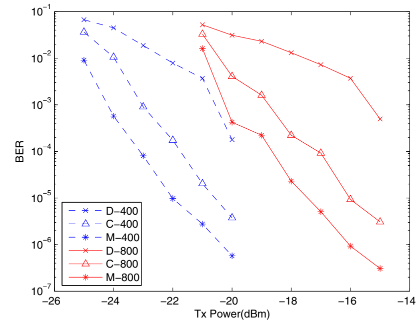


Fig. 11: Performance comparison of four-node cooperation.

TABLE II: BER vs. Tx Power under Various Transmission Rates.

BER	200Kbps			400Kbps			600Kbps			800Kbps		
TxPower	D	S	M	D	S	M	D	S	M	D	S	M
-27	8.07E-2	3.85E-2	1.81 E-2	-	-	-	-	-	-	-	-	-
-26	2.07E-2	1.85E-3	1.02E-3	-	-	-	-	-	-	-	-	-
-25	1.55E-2	9.88E-4	3.06E-4	6.70E-2	3.65E-2	9.06E-3	-	-	-	-	-	-
-24	6.13E-3	4.79E-4	4.07E-5	4.47E-2	1.06E-2	5.74E-4	-	-	-	-	-	-
-23	1.97E-3	2.24E-4	1.02E-5	1.88E-2	9.06E-4	8.03E-5	6.20E-2	3.28E-2	1.61E-2	-	-	-
-22	8.26E-4	1.06E-4	9.32E-7	7.87E-3	1.74E-4	9.76E-6	2.10 E-2	2.08E-3	7.22E-4	-	-	-
-21	3.49E-4	2.07E-6	3.05E-7	3.71E-3	2.03E-5	2.56E-6	1.30E-2	6.10E-4	3.22E-4	5.20E-2	3.28E-2	1.61E-2
-20	-	-	-	1.79E-4	3.76E-6	5.62E-7	7.12E-3	1.22E-4	2.30E-5	3.10E-2	4.08E-3	4.22E-4
-19	-	-	-	-	-	-	5.23E-3	6.22E-5	9.07E-6	2.30E-2	1.61E-3	2.22E-4
-18	-	-	-	-	-	-	1.70E-3	8.30E-6	7.29E-7	1.31E-2	2.22E-4	2.30E-5
-17	-	-	-	-	-	-	5.97E-4	2.07E-6	3.18E-7	7.23E-3	9.22E-5	5.07E-6
-16	-	-	-	-	-	-	-	-	-	3.70E-3	9.30E-6	8.80E-7
-15	-	-	-	-	-	-	-	-	-	4.97E-4	3.07E-6	3.26E-7

means that to achieve the same BER, cooperative transmission can save energy.

2) Multi-Relay Cooperation: For multi-relay cooperative transmission experiment, we set the scenario as shown in Figure 8.(c): there are totally four nodes in the experiment, one sender, one receiver and two relays. All the four nodes are located in a straight line with the sender fixed in one end and the receiver fixed in the other end. Two relays located next to each other in the middle point of the other two nodes. The distances from the relays to both sender and receiver are about 1.5 meter. The other setup is similar with the three node cooperation experiment. The only difference is that in the second timeslot, two relays forward the packet synchronously with the destination decode the super positioned signal directly. Figure 11 shows the BER performance under various transmission power of different transmission schemes. Results for direct transmission, single-relay cooperation and multi-relay cooperation is denoted by D, S and M respectively. The multi-relay cooperation shows significant performance gain in terms of BER compared with sing-relay cooperation, as multiple path is provided and higher level of diversity is exploited. For easy reading, only two levels of rate is shown in the figure, the results for other levels of rate can be found in Table II.

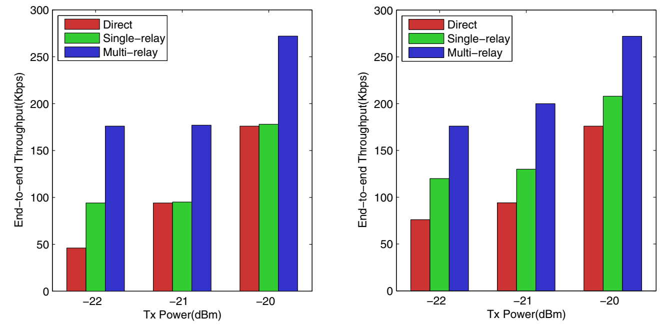
B. MAC Layer Performance Evaluation

In this subsection, we focus on the end-to-end throughput and delay performance from the MAC layer point of view. We propose a slotted MAC protocol with relay and receiver synchronized to the sender and feedback of channel status from the relay and receiver. One of the key features of our MAC protocol is its flexibility to switch among different transmission modes: direct transmission, single-relay cooperation and multiple-relay cooperation.

The control message is sent periodically, which is used by the relay and the receiver for synchronization upon receiving this message. The transmission mode indicator and the number of packets information is also included in the control message, which enable the relay and receiver switch the mode accordingly. Between consecutive control messages, the data packet is sent back-to-back for direct transmission. For the relay transmission, the sender and the relay alternatively send packet to the receiver. The receiver sends back ACK message periodically to indicate correct decoding.

Figure 12.(a) shows throughput of different transmission mode under various transmission power. Given certain maximal transmission power of both sender and relay, the maximum bit rate to guarantee that BER is smaller than 10^{-4} is selected, according to the link layer BER results under various bit rate (shown in Table II). The result shows that although there are some throughput gain of multi-relay cooperation compared with direct transmission, the single-relay cooperation can not always achieve throughput gain. This is because in multi-rate networks, there are only finite rate levels, if the interval between neighboring rate levels is too large, BER benefit can not always increase the data rate to be a higher level. If fine rate level interval is used e.g., 50Kbps between neighboring rate levels instead of 200Kbps, the throughput gain of both single-relay and multi-relay cooperation becomes significant as shown in Figure 12.(b).

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.



(a) Throughput under large rate level interval.

(b) Throughput under fine rate level interval.

Fig. 12: End-to-end throughput gain comparison.

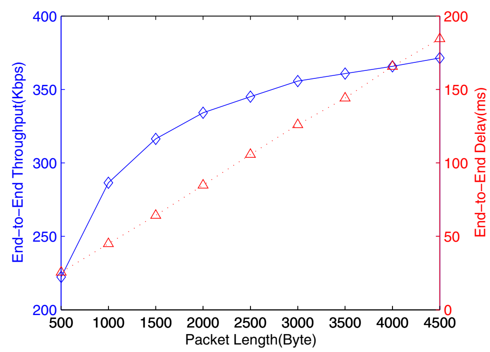


Fig. 13: Throughput and delay under various packet lengths.

It is also found that packet length effects the end-to-end throughput and delay significantly. Fix the transmission rate to be 400Kbps, as shown in Figure 13, both throughput and delay of the single relay cooperation increases while the packet length increases. The trend is the same for multi-relay cooperation. The throughput increases with packet length because that there is a switching time in relay node from receiving to transmit, which varies from 3.8 to 4.5 ms. With longer packet length, switching overhead per bit is reduced, which increased the throughput. However, as the receiver need to wait more time to combine signals together, the delay is increased with increasing packet length.

VII. Conclusion

In this paper, to evaluate the performance of cooperative communication schemes in real wireless environment, we build a testbed based on GNU Radio and USRP software-defined radio. Single-relay cooperation and multi-relay cooperation is implemented in the testbed. Some key techniques to do maximum ratio combine and synchronized transmission among multiple nodes in this testbed are provided. Extensive experiment is conducted and the result shows that cooperative communication significantly enhances the transmission reliability by exploiting spatial and user diversity.

Based on our cooperative testbed built in this paper, a lot of future works can be done. One direction is that more cooperative schemes such as amplify-and-forward, coded co-operation, can be implemented to investigate their link layer BER performance. We implement cooperative schemes based on GMSK modulation at this stage, in future work, we can also base on OFDM modulation to implement cooperative schemes, which can provide higher spectrum efficiency and easy to conduct channel estimation. Another direction is that more novel adaptive cooperative MAC protocols can be proposed to discuss the relay selection and rate adaptation issues. All these protocols can be implemented and evaluated in our testbed to verify its end-to-end throughput performance in real wireless environment. Moreover, various cooperative communication-aware routing protocols can also be verified based on this platform. No matter along which direction, our paper provides the basic functionality which is necessary for all cooperative scheme implementation.

VIII. Acknowledgement

This research was supported in part by Hong Kong RGC Grants HKUST 623209, 622508, N_HKUST609/07, and Huawei-HKUST joint lab.

References

[1] T. Cover and A. Gamal, "Capacity theorems for the relay channel," *IEEE Transactions on Information Theory*, vol. 25, no. 5, pp. 572–584, 1979.
[2] A. Bletsas, A. Khisti, D. Reed, and A. Lippman, "A simple cooperative diversity method based on network path selection," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 3, pp. 659–672, 2006.
[3] J. Laneman, G. Wornell, and D. Tse, "An efficient protocol for realizing cooperative diversity inwireless networks," in *IEEE ISIT*, 2001.
[4] T. Hunter, S. Sanayei, and A. Nosratinia, "Outage analysis of coded cooperation," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 375–391, 2006.
[5] G. Kramer, M. Gastpar, and P. Gupta, "Cooperative strategies and capacity theorems for relay networks," *IEEE Transactions on Information Theory*, vol. 51, no. 9, pp. 3037–3063, 2005.
[6] J. Laneman, D. Tse, and G. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3062–3080, 2004.
[7] A. Host-Madsen, "Capacity bounds for cooperative diversity," *IEEE Transactions on Information theory*, vol. 52, no. 4, pp. 1522–1544, 2006.
[8] T. Korakis, Z. Tao, S. Makda, B. Gitelman, and S. Panwar, "To Serve is to Receive Implications of Cooperation in a Real Environment," *Proceedings of Networking 2007*.
[9] Z. Per, M. Christos, S. Aris, *et al.*, "Experimental Investigation of Cooperative Schemes on a Real-Time DSP-Based Testbed," *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, 2009.
[10] "Gnu radio," `http://www.gnu.org/software/gnuradio`.
[11] "Usrp family brochure," `http://www.ettus.com/downloads/er_broch_trifold_v5b.pdf`.
[12] A. Bletsas and A. Lippman, "Implementing cooperative diversity antenna arrays with commodity hardware," *IEEE Communications Magazine*, vol. 44, no. 12, pp. 33–40, 2006.
[13] "Building a cooperative communications system," `http://warp.rice.edu/trac/attachment/wiki/JSAC_CooperativeComm/Files/Rice_JSAC_CooperativeComm.pdf`.
[14] "Wireless open-access research platform," `http://warp.rice.edu/index`.
[15] S. Katti and D. Katabi, "Embracing wireless interference: Analog network coding," *ACM Sigcomm*, 2007.
[16] S. Gollakota and D. Katabi, "Zigzag decoding: Combating hidden terminals in wireless networks," *ACM Sigcomm*, 2008.
[17] S. Katti and H. Balakrishnan, "Symbol-level network coding for wireless mesh networks," *ACM Sigcomm*, 2008.
[18] W. Kim, O. Khan, and K. e. Truong, "An experimental evaluation of rate adaptation for multi-antenna systems," in *IEEE INFOCOM*, 2009.
[19] G. Bradford, "A framework for implementation and evaluation of cooperative diversity in software-defined radio," Ph.D. dissertation, University of Notre Dame, 2008.
[20] "Osa 5200b gps clock," `http://www.oscilloquartz.com/index.php?pageid=91`.