A Demonstration of Video over a Cooperative PHY layer Protocol

Shashi Singh[†], Ehtesham Siddiqui[◇], Thanasis Korakis[†], Pei Lu[†], Shivendra Panwar[†]

[†]Department of Electrical and Computer Engineering
Polytechnic Institute of NYU, Brooklyn, NY
[◇] Department of Computer Science and Engineering, IIT Kanpur, Kanpur, India

E-mail: singh_shashiraj@yahoo.com, akhtariitkan@gmail.com, korakis@poly.edu, peilu@gmail.com, panwar@catt.poly.edu

***Abstract*—*Cooperative* communication is a promising approach to improve the reliability of a received signal at the physical layer. Cooperating nodes create a virtual MIMO system that provides spatial diversity even though the nodes have a single antenna. Therefore, tremendous improvements in system capacity and delay can be achieved. An implementation approach has been pursued in this demonstration to confirm the viability and efficacy of cooperation at the physical layer. The implemented cooperative physical layer scheme is called *CoopPHY* and is based on *Cooperative Coding*. In this technique, the source node transmits only a portion of its encoded data bits based on channel conditions among cooperating nodes. Helper nodes transmit the rest of the encoded bits. The destination node, finally, combines and decodes signals received via different channels and therefore increases the decoding reliability.In this demo, a Matlab video clip is streamed from a server to a client using sequentially the CoopPHY scheme and a non-cooperative scheme. The CoopPHY protocol delivers a smooth user experience, while the video play-out over the non-cooperative scheme shows frequent distortions. The demo clearly demonstrates the performance superiority of the implemented cooperative PHY layer scheme, as compared to the traditional transmission techniques that are based on a single hop transmission.**

Lab's website: *http://witestlab.poly.edu/*

I. The Cooperative PHY Protocol(CoopPHY)

The implemented cooperative PHY layer scheme is called *CoopPHY* and is based on *Cooperative Coding*, a technique that achieves diversity by using distributed FEC coding [1]. The basic functionality of the protocol is illustrated in Figure 1 and summarized as follows.

When a wireless node experiences a bad channel it uses cooperative communication by recruiting a relay station. The relay, which we call *helper* in this protocol, has a good channel with both the source and destination stations. As shown in the Figure, source node encodes a packet of length *k* bits using a block of $n = n1+n2$ bits. This is done by using convolutional coding with a coding rate *R*=*k*/*n*, where $R < 1$. Based on the channel condition among the nodes(source, helper and destination), the source removes $n2$ bits out of the original n coding bits by using a *puncturing* technique. After this, the source node transmits the remaining $n1$ bits. The helper node receives these n1 bits and decodes with high probability due to the good channel between the source and itself. If the decoded data is correct, as determined by CRC, helper re-encodes the data using $n2$ bits and transmits. Destination recombines the two transmissions and decodes them. Using this technique the decoding of the frame is more robust, a fact that can lead to lower PER for the same transmission rate or to higher transmission rates for the same PER.

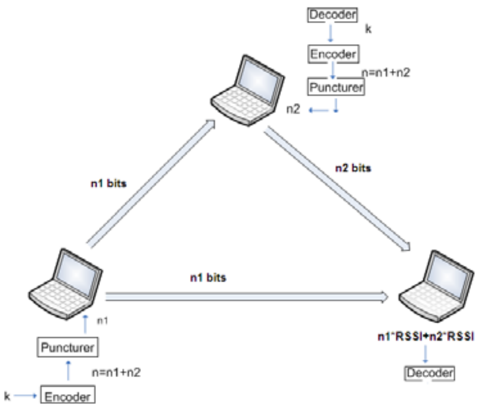

Fig. 1: Cooperation at PHY layer.

II. Implementation of the PHY Layer Cooperative Scheme

The implementation of the CoopPHY protocol in this demo is done on the WARP platform [2]. For the upon the physical layer design we used the *WARPLAB* framework [3] In this framework, the PHY layer has been designed as a collection of matlab files. Main features of the design include *convolutional coding* with a coding rate equal to $1/2$ and a memory between 3 and 7, *rate compatible punctured codes* for the selection of the coded bits that will be transmitted by the source and the helper, an enhanced *Viterbi decoder system that is able to decode using RSSI as weights for each coded signal*.

III. Demonstration of Video over CoopPHY

A. Demo Configuration

The demo consists of 1 laptop and 3 WARP nodes(source, helper and destination), WARP nodes are connected to the laptop through an Ethernet cable. The basic positioning of the nodes is outlined in Figure 2. The destination node is kept behind an obstruction such that it is not in line-of-sight with the source or the helper node. Source and helper nodes are in line-of-sight with each other.

The 3 WARP nodes act as a transceiver for the source, helper and destination. The logical part of the three stations resides in the laptop, in the form of matlab files.

B. Demo Description

The transmission of a matlab video clip is considered in the described testbed. Source station continuously transmits frames of a video clip over the air. The destination station receives the video and it plays it in matlab. The demo consists of three sequential phases, as outline in Table I. Once all three phases are completed the cycle starts from the beginning. To

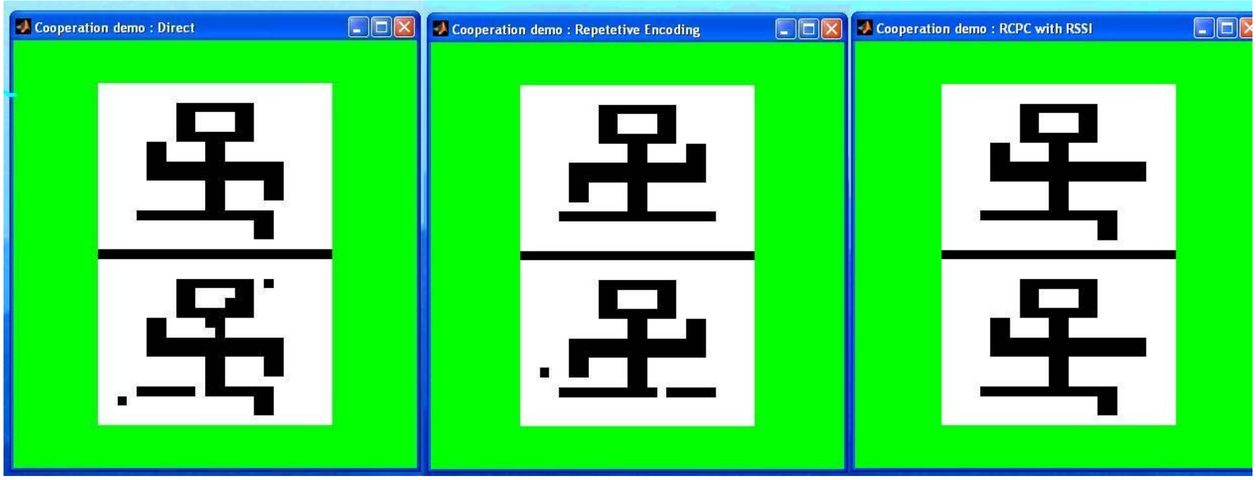
Cooperation demo : Direct
Cooperation demo : Repetetive Encoding
Cooperation demo : RCPC with RSSI

Fig. 3: Video Quality Comparison: A Snapshot.

TABLE I: Three Phases in the Demonstration

Phase 1	Direct mode: Non-cooperative network direct transmission received by the destination. The quality of the video is bad.
Phase 2	Cooperative mode w/o the use of RSSI weights. Two transmissions are received at the destination RSSI is not used in decoding. The quality of the video is good.
Phase 3	Cooperative mode with the use of RSSI. Two transmissions received at the destination. RSSI weighting is used in decoding. Best quality of video.

automate the phase transition, a matlab script is written that switches sequentially from mode to mode every 30 secs.

As described in Table I, the user perception is poor for video transmission in the non-cooperative mode, as noticeable distortions occur frequently. Meanwhile, the video is smooth, when it is received over a cooperative PHY network. Between the two cooperative modes, the one that use RSSI weighting in the decoding process gives the best video quality. Figure 3 provides a snapshot of the video taken at the destination for non-cooperative mode (sub-figure in the left), the cooperative PHY mode scheme without the use of the RSSI weighting (sub-figure in the middle) and the cooperative PHY scheme with the RSSI weighting technique (sub-figure in the right). In each sub-figure, the image in the top is the one that is transmitted in the source and the image in the bottom is the one that is received by the destination. The comparison of the three images in the bottom is typical and reveals the substantial improvement in the video quality that CoopPHY can deliver. A video of the demo can be found in the project's site [4].

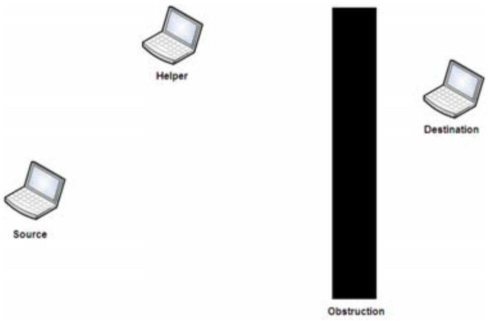

Fig. 2: Nodes setup.

C. A Graphical User Interface: CoopGUI

To facilitate the demo, a graphical user interface (GUI) called *CoopGUI* has also been developed. Based on MATLAB, CoopGUI provides the user interface to run and stop demo. It also provides an option to run only non-cooperative mode or cooperative mode. A snapshot of the transmitter's CoopGUI is depicted in Figure 4.

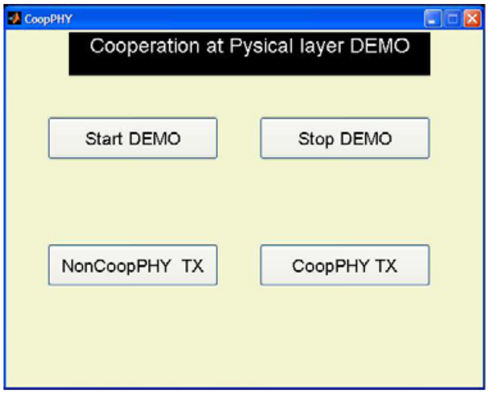

Fig. 4: The Transmitter's GUI.

D. Procedural Details

The demo needs a table to place the laptop and the three WARP nodes on. It needs power support for all the four devices. The setup time is 15 minutes. The demo is eligible for the student demo competition. The lead students are Shashi Raj Singh from Polytechnic Institute of NYU and Ehtesham Akhtar Siddiqui from IIT Kanpur.

References

[1] A. Stefanov and E. Erkip, "Cooperative Coding for Wireless Networks," *IEEE Transactions on Communications*, vol. 52, no. 9, pp. 1470–1476, 2004.
[2] "WARP," *http://www.warp.rice.edu*, vol. 38, pp. 531 – 548, March 2002.
[3] "WARPLAB," *http://warp.rice.edu/trac/wiki/WARPLab*.
[4] "WITESTLAB," *http://witestlab.poly.edu/*.