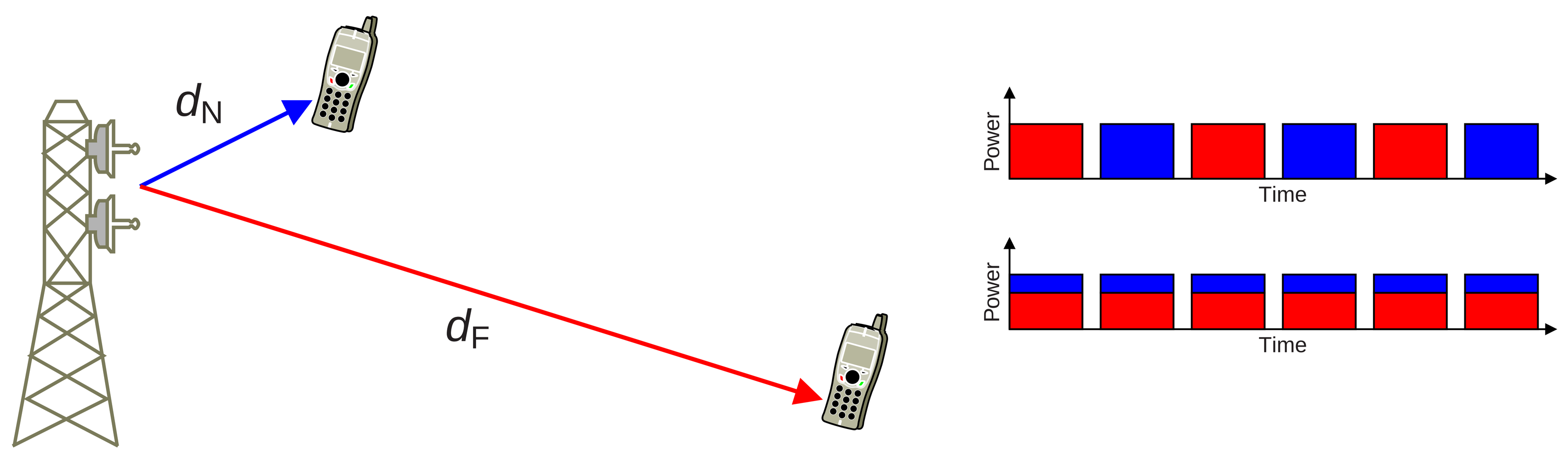
Implementation and Experimental Results of Superposition Coding on Software Radio

R. K. Ganti, Z. Gong, M. Haenggi, C. Lee, S. Srinivasa, D. Tisza, S. Vanka and P. Vizi
Department of Electrical Engineering, University of Notre Dame

RF Alliance Conference, Purdue University, April 5-6, 2010

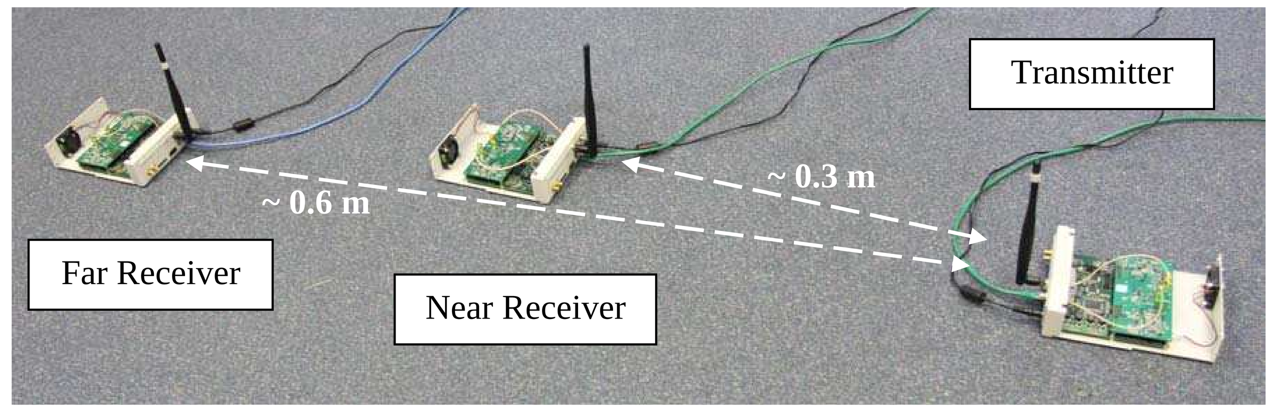
UNIVERSITY OF NOTRE DAME

Overview of Superposition Coding



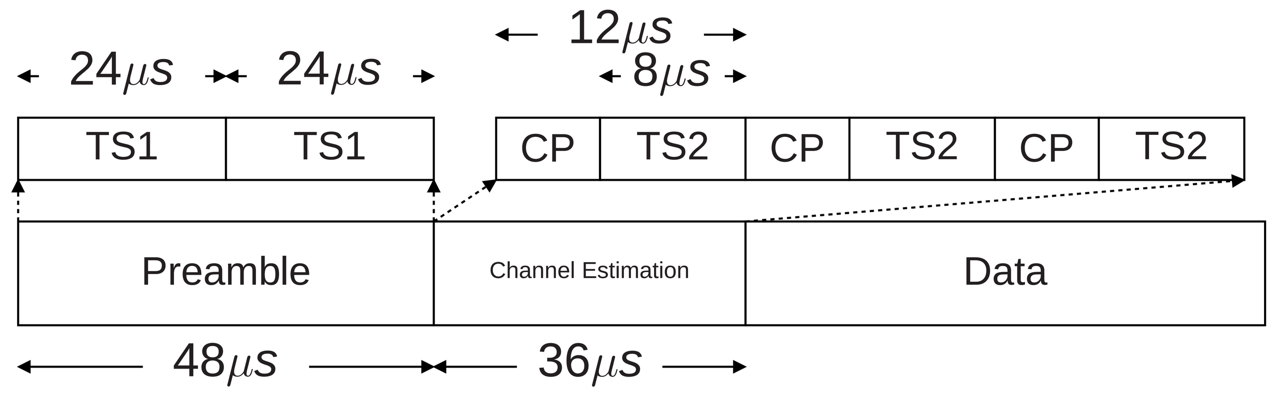
- The base station (BS) wishes to communicate with two users N (near) and F (far).
- Suppose that N is closer to the BS, while F is located at the cell edge ($d_N \ll d_F$).
- This scenario is modeled as a *stochastically degraded* Gaussian broadcast channel.
- The capacity region is known, and is achievable employing *Superposition Coding* and *Successive Decoding*.

Implementation of Superposition Coding



Implementation set-up.

- Hardware: USRP 1 (Analog and RF front-end).
- Software: GNU Radio (ver. 10923) and its built-in libraries.



Physical frame structure.

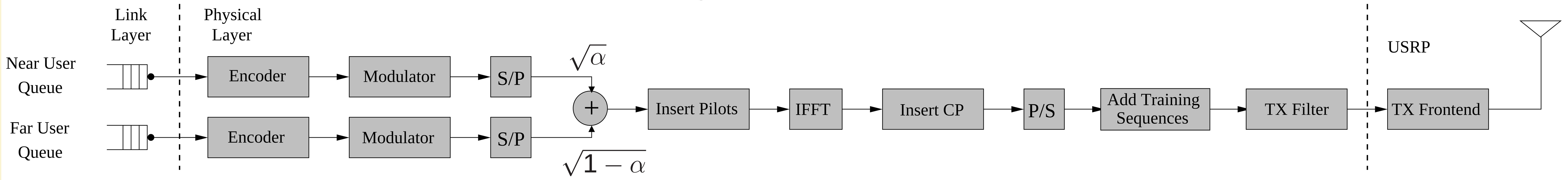
- TS1: Training sequence 1.
- TS2: Training sequence 2.
- CP: Cyclic Prefix.

System Parameters

Center Frequency	903 MHz
System Bandwidth	1 MHz
Payload Size	0.5kBytes (incl. 4-Byte CRC)
Transmission Scheme	16-tone OFDM
Data Tones	8
Pilot Tones	4
Null Tones	4
Cyclic Prefix Length	4μs
Far User Modulation	BPSK
Near User Modulation	BPSK
Gen. Poly. for Conv. Code	[133, 171]
Far User Code Rate	1/2
Near User Code Rate	1/2
Power allocation α	0.8

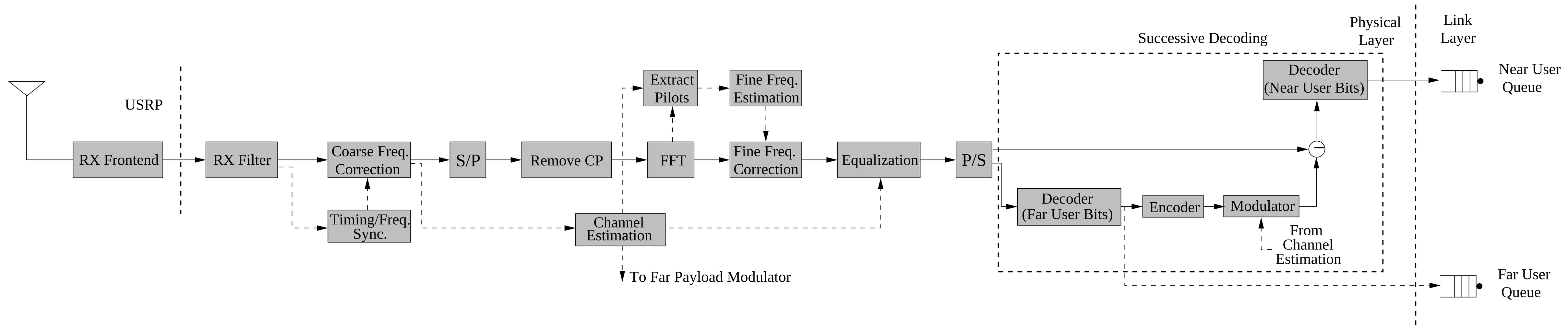
Transmitter and Receiver Operation

Block diagram of the transmitter:



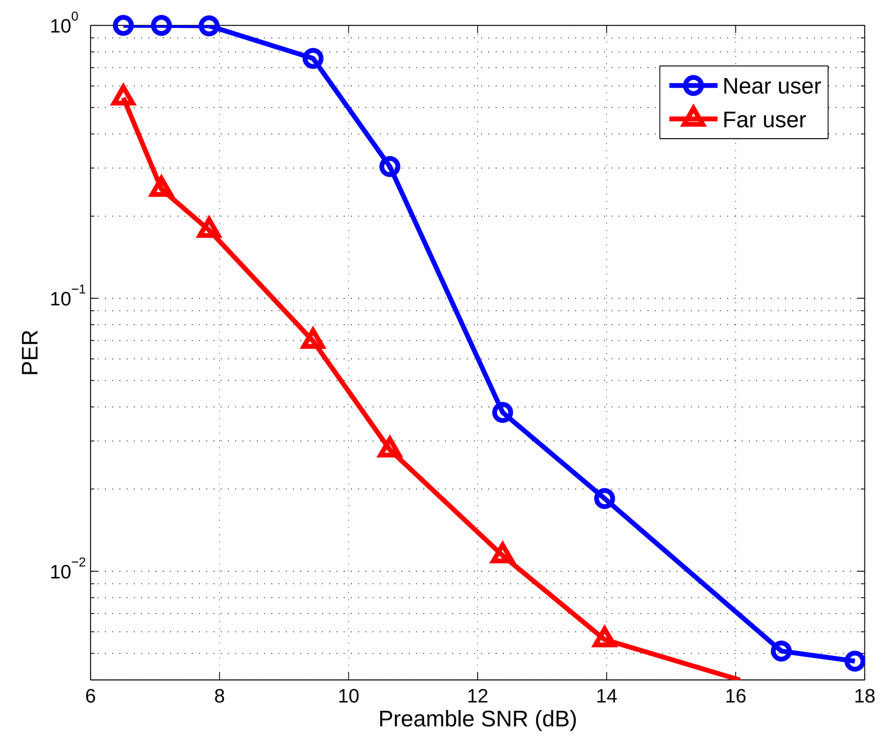
- The payloads (near and far user data in bits) are provided to the physical layer by the link layer.
- The signals are weighted by the fraction of power allocation factors ($\sqrt{1-\alpha}$ for the near user signal, and $\sqrt{\alpha}$ for the far user signal), added and transmitted.

Block diagram of the receiver:



- Near user first decodes the far user bits and then employs successive decoding to decode its own message.
- Far user decodes its payload by simply treating the near user signal as interference.

PER versus SNR



PER versus SNR for both the near and far users for $\alpha = 0.8$.

The preamble SNR is 3dB above the payload SNR.

Lessons Learned

- **Packetized system support urgently necessary for GNU Radio.** The GNU Radio scheduler was designed for a **flow-based framework**, i.e., each block operates on a stream of data rather than packets of data. This makes the design of a frame-based system quite challenging.
- **Complex system is easier to implement in one signal processing block.** Implementing **feedback loops** between signal processing blocks is cumbersome. From our experience, implementing all the functions in one GNU Radio block significantly simplifies the design.
- **Interpolation has to be done in software.** The effective bandwidth of the USRP1 (the first generation USRP) is much smaller than that set by the user. The cause of this problem lies in the highly **non-ideal transmit path** implementation of the USRP1. We observe that the 3*dB* bandwidth is about 450 kHz for an interpolation factor of 128. The solution is to implement a software-based filter and do part of the interpolation in software.

Conclusions

- We have presented a design that leverages the flexibility of the GNU Radio/USRP platform to implement physical layer of a communication system that uses superposition coding.
- The performance of the implementation was measured using packet error rates for each of the (near and far user) streams.
- We found that the limitations of the USRP hardware have implications for the design of training sequences, and the software environment imposes constraints on the degree of modularity allowed in the system implementation.

Transmitter Non-Ideality