Spinal code
Jonathan Perry, Peter A. Iannucci, Kermin E. Fleming, Hari Balakrishnan, Devavrat Shah, MIT.

The paper described a rateless code construction based on random hash function over k-bit blocks, and an efficient “bubble” decoder that achieves near-optimal channel capacity using reasonable computation.

In class, we discussed the relationship between using rateless code and having channel estimates in today’s cellular wireless channel implementation. As MIMO becomes prevalent, channel quality is constantly being estimated, hence it’s tempting to use a rated code. However, compared with other rated code such as LDPC/Turbo, Spinal code still has the benefit of easy parallelizable decoding.

The spinal code is in fact combining coding with modulating, and is suitable for integrating with any symbol set. This is due to the fact that spinal code produces pseudo-random output, making mapping to any constellation suitable. Also, due to hashing, an error in decoded bit becomes obvious in encoded bits/symbols as all subsequent bits/symbols are changed.

We also discussed the sending model of spinal code (sender keeps sending until receiver acks successful decode) is different from many of today’s wireless implementation, especially cellular, where sending time slots are scheduled ahead of time; also, the wireless medium is often half-duplex, so switching the direction to send an ack require switching the physical layer. In this case, a spinal code sender must switch back to listening mode after sending every pass to check for acks. Recent trend in cellular wireless networks has enabled more and more simultaneous uplink and downlink transmissions, making it easier to implement acks. An alternate coding mode where the sender always sends fixed length signals but the receiver receives partial messages (depending on signal quality) is called “superposition coding”, in contrast to rateless code.

We also note that the choice of efficient hash function in spinal code implementation is important for achieving high performance, and it is not necessary to use cryptographically secure hash function as long as the channel is merely noisy, not adversarial. We only require the hash function to exhibit good randomness properties, such as avalanche effect. However, it is possible for an adversarial attacker controlling the channel to cause the decoder to decode into a chosen plaintext message, even with the correct CRC, as both the hash function, RNG, and CRC are not cryptographically resistant to attacks.

Given the parameter choice of k-bit per block generating v-bit state, and sending c-bit per symbol, information theoretically, we need at least k/c passes to gather enough information for the first decoding attempt (without noise). In the evaluation, the larger gap to Shannon limit at the higher-SNR region is caused by a discretization effect of the required number of passes. At lower SNR, we require many number of passes, and the discretization effect goes away.

We also note that it is interesting to investigate the energy consumption of the bubble decoder, as it requires a lot of computation and might drain the battery of mobile devices.