Interference Management via Capacity-Achieving Codes for the Deterministic Broadcast Channel

Todd P. Coleman, Emin Martinian, Michelle Effros, and Muriel Médard
colemant@mit.edu, emin@alum.mit.edu, effros@caltech.edu, medard@mit.edu
MIT, Mitsubishi Electronic Research Labs, Caltech, MIT

***Abstract*— We motivate the consideration of deterministic broadcast channel coding as an interference management technique in wireless scenarios. We address practical coding strategies for such channels and discuss two approaches. The first relies upon enumerative source coding and can be applied for any deterministic broadcast channel problem as the first step in pipelined encoding for vertex rates. The second approach addresses a wireless interference management scenario and is a complete, practical, capacity-achieving strategy that dualizes the Luby Transform code construction and encoding/decoding algorithms. This results in the first practical, non-trivial, capacity achieving code construction for the deterministic broadcast channel.**

I. Introduction

One major issue in wireless networks is interference. At a high level, we understand how to deal with noise relatively well, but our understanding of the interaction between multiple transmitters and receivers is still somewhat limited. As a result, most network designs attempt to suppress these interactions by making the different users orthogonal or performing successive cancellation. As we argue in the following paper, however, when properly managed, these interactions can actually result in a net benefit to the system.

Consider the wireless communication scenario shown in Figure 1 where two users both receive signals from a pair of cooperating transmitters. Intuitively, the interference to Rx 1 from Tx 2 is negligible since the signal from Tx 2 experiences much higher path loss than Tx 1. In contrast, Rx 2 receives potentially interfering signals from both Tx 1 and Tx 2. Similar situations can arise in wireless networks employing multi-hop transmission as illustrated in Figure 2. How can we model such scenarios to understand the effect of such interference and develop efficient communication schemes? One approach is to consider a Gaussian broadcast channel model where the transmitter has perfect channel side information (CSI) describing the propagation parameters. For such models, the capacity region [1] is obtained using Costa's idea of writing on dirty paper [2].

Unfortunately, this approach requires perfect knowledge of the channel which is sometimes unrealistic. Thus it is not clear to what extent the so-called dirty paper coding ideas apply to non-coherent communication or to relay networks like the one in Figure 2, which employ distributed transmitter cooperation. Furthermore, even when perfect channel state information is available, no practical coding scheme is known that achieves capacity. Specifically, to our knowledge, the best known dirty paper coding systems are around a decibel away from capacity

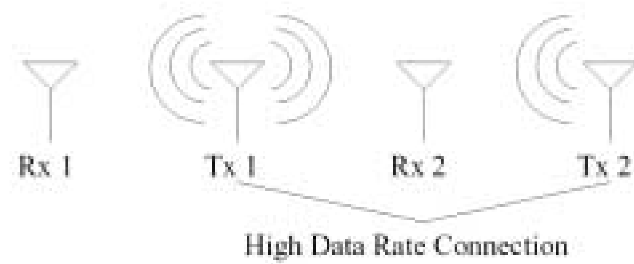

Fig. 1. A wireless downlink communication scenario. One user, Rx 1, receives only the signal from the first transmitter, Tx 1, while the second user, Rx 2, receives signals from both transmitters Tx 1 and Tx 2. If the two transmitters are connected by a high data rate link such as a land-line for terrestrial transmitters or a laser communication link for satellites, transmitter cooperation can increase the communication rate.

[3], [4]. While a couple of decibels may seem like a small gap to quibble over, at low signal-to-noise ratios which are common in certain types of wireless networks, a couple of decibels may correspond to a large fraction of the transmitted power or rate. For multi-antenna channels, the gap to capacity may be larger.

The existence of this gap to capacity may seem surprising in light of the spectacular success of turbo codes [5], low density parity check codes (LDPCs) [6] and other codes on graphs in approaching capacity for single user channels. Intuitively, the gap to capacity for dirty paper coding is caused by the lack of efficient codes for the shaping/binning required in the random coding arguments for broadcast channels.

Consequently, we approach the wireless network communication problem in Figures 1 and 2 from a different perspective. Since dealing with noise via coding is fairly well understood at this point, we focus purely on interference issues by considering a deterministic broadcast channel (DBC). This allows us to develop clearer insights into the effects of interference and appropriate coding systems.

Specifically, we consider a simple deterministic broadcast channel model where we are able to evaluate the Marton-Pinsker capacity region [7], [8], [9] and develop codes that achieve it. Coding at vertices allows for significant complexity reduction at the encoder by performing a pipelined strategy. Since rate-splitting can be applied to transform any point in the DBC to a vertex [10], we constrain our focus to vertex strategies. Inspired by enumerative source coding [11], we introduce encoding and decoding strategies that apply to the construction

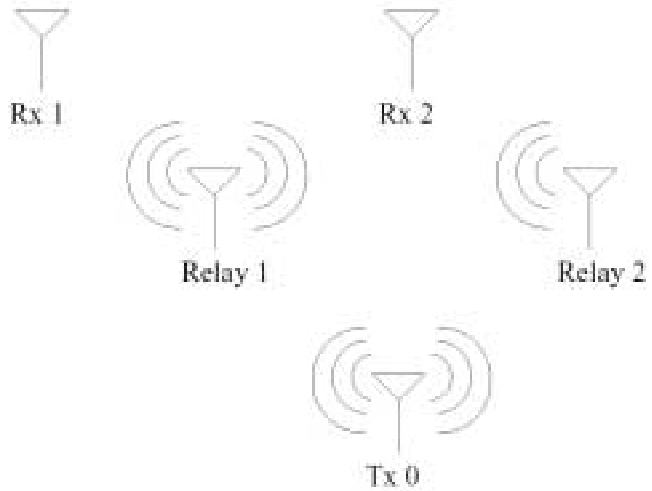

Fig. 2. A wireless relay communication scenario. A single transmitter, Tx 0, sends a message intended for two receivers Rx 1 and Rx 2 via multi-hop transmission. The two intermediate relay nodes each decode the message, re-encode, and transmit to the ultimate destination. Rx 1 sees only the signal from Relay 1, while Rx 2 receives potentially interfering signals from both relays.

of the first output sequence in the vertex encoding pipeline of an arbitrary DBC. The code construction to achieve capacity for the interference scenario dualizes the Luby Transform [12] code construction and encoding/decoding algorithms.

A. Interference Management Via Deterministic Broadcast Channels

We now consider a DBC model based on Figures 1 and 2 with a pair of binary input symbols $X = [X^1, X^2]'$ and two outputs Y^1, Y^2. Intuitively, X^i corresponds to the channel input for Tx i or Relay i in Figures 1 and 2. If the two inputs are the same, then they do not destructively interfere. But if they differ then receiver 1 is unaffected and receives X^1 correctly while receiver 2 suffers destructive interference in the sense that it can not determine X^2. We model this scenario via the DBC in (1):

$$\text{if } X^1 = X^2 \quad \text{then} \quad Y^1 = X^1 \text{ and } Y^2 = X^2 \tag{1a}$$
$$\text{if } X^1 \neq X^2 \quad \text{then} \quad Y^1 = X^1 \text{ and } Y^2 = * \tag{1b}$$

where $*$ means 'erasure'. Thus we have the following input-output relationship:

(X^1, X^2)	Y^1	Y^2
$(-1, -1)$	-1	-1
$(-1, 1)$	-1	$*$
$(1, -1)$	1	$*$
$(1, 1)$	1	1

The channel in (1) can model a variety of physical scenarios. Perhaps the simplest is binary phase shift keying (BPSK) with additive combining. For this model, the channel inputs are $X^i = \pm 1$ with $Y^1 = X^1$ for the receiver without interference and additive interference corresponding to $Y^2 = X^1 + X^2$ for the other receiver. Thus the $*$ output in (1b) represents the case where $X^1 = -X^2$ resulting in a received signal of $Y^2 = 0$. Equation (1) can also represent non-coherent modulation such as frequency shift keying (FSK). In an FSK model, each transmitter sends either on frequency f_0 or f_1 corresponding to $X^i = 0/1$. If the two transmitted signals both equal t, then both receivers see a signal on frequency f_t and decode correctly. If the two transmitted signals are opposite, the first receiver sees no interference and decodes correctly while the second receiver observes signals on both frequencies corresponding to an erasure.

II. BACKGROUND ON THE DBC

The deterministic broadcast channel has one sender and multiple receivers. The sender combines the M independent messages $\{m_j \in \{1, \ldots, 2^{nR_j}\}\}_{j=1}^{M}$ to be sent to each receiver into a single length-n string $\underline{x} = (x_1, \ldots, x_n)'$, where $x_i \in \mathcal{X}$. At receiver j each symbol $y_i^j \in \mathcal{Y}_j$ is a deterministic function of x_i, i.e. $y_i^j = f_j(x_i)$. The jth decoder attempts to reconstruct m_j, i.e. $\hat{m}_j = d_j(\underline{y}^j)$. A memoryless probability distribution $P(X)$ on x, combined with $f_1, \ldots, f_M$, induces a memoryless joint distribution $P(Y^1, \ldots, Y^M)$ on $\{y^1, \ldots, y^M\}$. For a fixed memoryless $P(X)$ the set of all achievable rates $\mathcal{R}\left[P(X); f_1, f_2, \ldots, f_M\right]$ is given by [7], [8], [9]

$$\left\{ \underline{R} \in \mathbb{R}_+^M \;\middle|\; \sum_{i \in S} R_i < H\left(Y(S)\right) \;\forall\, S \subseteq \{1, \ldots, M\} \right\}, \tag{2}$$

where $Y(S) = \{Y^j, j \in S\}$. The capacity region of the DBC is given by

$$\mathcal{R}\left[f_1, f_2, \ldots, f_M\right] = \mathrm{cl}\left[\mathrm{CH}\left(\bigcup_{P(X)} \mathcal{R}\left[P(X); f_1, f_2, \ldots, f_M\right]\right)\right]$$

where cl denotes closure and CH denotes convex hull.

In this paper, we focus on the two-receiver case and postpone the M-receiver problem to future work.

A. Binning as an Achievable Strategy

As in the Slepian-Wolf (SW) problem [13], binning has been discussed as a strategy to attain any achievable rate. For $j = 1, 2$, all possible output sequences $\underline{y}^j \in \mathcal{Y}_j^n$ are partitioned into a set of 2^{nR_j} bins. At receiver j, the DBC decoder observes $\underline{y}^j$ and specifies as its output the bin index associated with that sequence. The DBC encoder observes as its input a set of messages that specify, for each receiver, which bin the received sequence should lie in. The DBC encoder next searches within these bins for a (usually non-unique) set of jointly typical sequences $(\underline{y}^1, \underline{y}^2)$. Having selected the output of the channel, the encoder's final step is to choose an input sequence to produce this output. Specifically, for every $i \in \{1, \ldots, n\}$, the tuple (y_i^1, y_i^2) is used to select any $x_i \in \cap_{j=1}^2 f_j^{-1}(y_i^j)$.

B. Vertices: Successive Encoding

If we consider an encoding strategy with a memoryless probability distribution $P(X)$, then the set of achievable rates $\mathcal{R}[P(X); f_1, f_2]$ has *vertices* or *corner points* associated with expanding $H(Y^1, Y^2)$ into 2 terms by successive applications of the chain rule for entropy and assigning to each rate the unique corresponding term in the expansion. Transmitting at such rates allows for the joint search over all users' bins to be done successively. For example, consider communicating at the rate $(R_1, R_2) = \left(H(Y^1|Y^2), H(Y^2)\right)$:

- Encoding message m_1 at rate $R_1 = H(Y^1)$ can be done by simply searching in the bin of message m_1 for a typical $\underline{y}^1$ sequence. This is because there are 2^{nR_1} such bins, one for each message m_1 index, and there are asymptotically $2^{nH(Y^1)}$ typical $\underline{y}^1$ sequences.
- After successful encoding of m_1, encoding message m_2 at rate $R_2 = H(Y^2|Y^1)$ can be done by simply searching in the bin of message m_2 for a sequence $\underline{y}^2$ that allows for $(\underline{y}^1, \underline{y}^2)$ to be jointly typical. This is because there are 2^{nR_2} such bins, one for each message m_2 index, and there are asymptotically $2^{nH(Y^2|Y^1)}$ sequences $\underline{y}^2$ that allow for $(\underline{y}^1, \underline{y}^2)$ to be jointly typical.

Figure 3 illustrates the successive encoding mechanism.

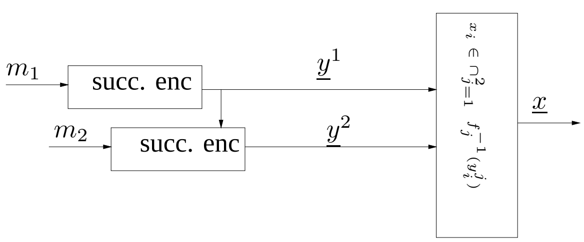

Fig. 3. Pipelined Encoder for Communicating at a Vertex Rate for the DBC.

III. Practical challenges for the DBC

We now discuss some practical issues for the deterministic broadcast channel that differ from other practically solvable multi-terminal binning problems (such as Slepian-Wolf [13]).

A. Shaping

Depending on the desired output joint distribution $P(Y^1, Y^2)$, the optimal input to the channel $P(X)$ need not be uniform. A shaping code must take uniform bits and map them to symbols with non-uniform probabilities. This operation is in some sense the dual of lossless compression systems which take non-equiprobable symbols and map them to uniform bits. Gallager [14, pp. 208-209] discusses one encoding approach using linear codes, but he also notes that the decoding process is prohibitively complex. We note that in such a situation, it is *unclear* how to traditionally use a linear code to map the inputs to outputs with low encoder and decoder complexity.

B. Binning

Another interesting coding question for this problem is the construction of low complexity, efficient binning codes. Specifically, in the random coding argument proving the capacity region of broadcast channels, a random codebook is generated according to the joint distribution $P(Y^1, Y^2)$. Each codeword is assigned a bin number for Rx 1 and a bin number for Rx 2. To send messages m_1 and m_2, jointly typical output codewords $(\underline{y}^1, \underline{y}^2)$ that lie in bin numbers (m_1, m_2) are selected.

Having selected the output of the channel, the DBC encoder's final step is to choose an input sequence to produce this output. Specifically, for every $i \in \{1, \dots, n\}$, the tuple $(y_i^1, y_i^2, \dots, y_i^M)$ is used to select any $x_i \in \cap_{j=1}^{M} f_j^{-1}(y_i^j)$. For all achievable rates, with high probability there will be *exponentially many* such jointly typical codewords. Thus the *key difference* in binning codes for deterministic broadcast channels vs. binning codes for other problems such as Slepian-Wolf coding is that there is only a single jointly typical pair in the latter while *many jointly typical pairs* exist in the former. This difference manifests itself in the difficulty of applying existing iterative coding ideas to the general (non-deterministic) broadcast problem. By focusing on the special deterministic channel, however, we develop the first non-trivial, low complexity, capacity-achieving DBC codes.

IV. Practical First-Stage Vertex Pipelined Encoding for the General DBC

We now consider communicating at a vertex rate and attempt to construct the first sequence $\underline{y}^j$ in the pipeline, which has a rate given by $R_j = H\left(Y^j\right)$. This uses an enumerative source coding [11] data compression strategy proposed by Cover, which involves the method of types [15]. Consider a typical sequence $\underline{y}$ we attempt to construct according to $P(Y)$. Define:

$$\begin{aligned}
\mathcal{M} &= \{0, 1, \dots 2^{nR} - 1\} \\
P_{\underline{y}} &= \left(\left\{ \frac{1}{n} \sum_{i=1}^{n} 1_{y_i = a} \right\}_{a \in \mathcal{Y}} \right) \text{ for } \underline{y} \in \mathcal{Y}^n \\
\mathcal{P}_n(\mathcal{Y}) &= \left\{ P_Y^n \in \mathcal{P}(\mathcal{Y}) : P_Y^n = P_{\underline{y}} \text{ for some } \underline{y} \in \mathcal{Y}^n \right\} \\
T(P_Y^n) &= \left\{ \underline{y} \in \mathcal{Y}^n : P_{\underline{y}} = P_Y^n \right\}
\end{aligned}$$

For a type $P_Y^n \in \mathcal{P}_n(\mathcal{Y})$, we construct our encoder $E : \mathcal{M} \to T(P_Y^n)$. Note that this constrains every codeword $\underline{y}$ to lie in the same type class $T(P_Y^n)$. For a target distribution $P(Y)$, we select P_Y^n such that

$$P_Y^n \in \arg \min_{\tilde{P} \in \mathcal{P}_n(\mathcal{Y})} \left| \tilde{P} - P(Y) \right|_1 .$$

Because $\mathbb{Q}$ is dense in $\mathbb{R}$, $P_Y^n \to P(Y)$. Moreover, since

$$\begin{aligned}
|T(P_Y^n)| &= \binom{n}{nP_Y^n(0)\; nP_Y^n(2)\; \dots\; nP_Y^n(|\mathcal{Y}_1| - 1)} \quad (3) \\
&= 2^{n[H(P_Y^n) - o(n)]}
\end{aligned}$$

where $o(n) \to 0$, we have that

$$\frac{1}{n} \log |T(P_Y^n)| \to H(Y).$$

We construct a lexicographic order on each $\underline{y} \in T(P_Y^n)$. Any two sequences $\underline{y}^1, \underline{y}^2 \in T(P_Y^n)$ can be compared order-wise by observing the first leftmost symbol j such that y_j^1 and y_j^2 differ. The order relation gives higher precedence to the sequence with larger value in the jth symbol.

The decoder $D : T(P_Y^n) \to \mathcal{M}$ maps a given $\underline{y} \in T(P_Y^n)$ to its position in the lexicographically ordered alphabet by performing combinatorial calculations by exploiting (3).

If we consider multiplication as a fixed cost operation, then encoding and decoding can be done in $O(n)$ time by saving in memory, throughout the process, previously counted operations and dividing or multiplying by at most $|\mathcal{Y}|$ numbers. The encoding and decoding process follows from [11].

V. A Capacity-Achieving Low-Complexity Solution for DBC Interference Management

We now discuss the DBC problem presented in I-A. Consider maximizing R_2, which is done by generating $\underline{y}^2$ according to the uniform distribution on $(-1, 1, *)$. Given $R_2 = H(Y^2) = \log_2(3)$, we see from the input-output relationship table in section I-A that to maximize $R_1 = H(Y^1|Y^2)$, we must select a $P(X)$ that maximizes $H(X) = H(Y^1, Y^2)$ subject to $P(-1,-1) = P(1,1) = \frac{1}{3}$. Thus $P(x)$ should be drawn according to the distribution:

(X^1, X^2)	$P(X^1, X^2)$
$(-1,-1)$	$\frac{1}{3}$
$(-1,1)$	$\frac{1}{6}$
$(1,-1)$	$\frac{1}{6}$
$(1,1)$	$\frac{1}{3}$

Since $R_2 = H(Y^2)$, we can construct $\underline{y}^2$ by using the enumerative source coding technique discussed in Section IV.

A. Coding to Construct $\underline{y}^1$

One general approach for using linear codes for binning in this problem is as follows. A code and its parity check matrix H are fixed before transmission begins. A sequence $\underline{y}^2$ is selected for Rx 2 based on the shaping code. To send a message, $\underline{v}$, to Rx 1, two conditions must be satisfied. The first is that $H \cdot y^1 = v$, which ensures that Rx 1 can decode the message v by looking at the bin index of $\underline{y}^1$. The second condition is that $\underline{y}^1$ and $\underline{y}^2$ are consistent, which is represented by the equation $T \cdot y^1 = T \cdot y^2$ where T is identity matrix with entries corresponding to the $*$ symbols in $\underline{y}^2$ set to 0. We can combine these equations into a single linear system using block matrices to get

$$[HT] \cdot y^1 = [vT \cdot y^2].$$

Thus a general linear code used for this problem needs a matrix inversion to determine $\underline{y}^1$ requiring $O(n^3)$ complexity. Ideally, we would like to use a low density parity check code or some other sparse graph code to reduce this complexity.

ERASURE-DECODE-BEC$(G, \underline{y})$
1. **While** $\underline{u}$ has at least one unrecovered sample **do**
2. **if** $\exists$ one unerased (check) i connected to exactly one neighbor u_j **then**
3. Recover u_j immediately and propagate it to any adjacent unerased checks i' via $y_{i'} = y_{i'} \oplus u_j$.
4. **else**
5. **return** FAIL
6. **end if**
7. **end while**
8. Set $\underline{u}$ to the values from the checks obtained from $\underline{y}$
9. **return** $\underline{u}$

Duals of LT Codes We now exploit the structure of this problem and its similarity to binary erasure quantization using codes on graphs [16]. In that setting, a sequence of symbols $\underline{y} \in \{0, 1, *\}^n$ is given to an encoder which decodes to $\underline{x} \in C$ for some binary linear code C such that $\underline{x}$ agrees with $\underline{y}$ in non-erased positions. There are an exponential number of such $\underline{x}$'s for any typical $\underline{y}$, just as in our case. By dualizing capacity-achieving parity-check graphical representations of linear codes for the binary erasure channel (BEC), the authors construct a rate-distortion optimal generator-form graphical representation of the linear code for binary erasure quantization. The dual quantization algorithm discussed in [16] fails if and only if the analogous BEC decoding algorithm fails. One slight difference in our setting is that there is an extra constraint that must be satisfied: if H is the parity-check matrix for C, then we must have $H\underline{x} = \underline{s}$ where $\underline{s}$ is the message bin index. Thus mapping from a parity-check representation to a generator representation will not apply here, because the generator matrix for any code produces codewords $\underline{x}$ that lie in C, which means that $H\underline{x} = \underline{0}$. Moreover, attempting to dualize a generator representation that has a graphical representation like that of an LDPC will provably fail: any representation with a constant fraction of nodes with bounded degree will have a non-negligible probability of encoding failure [16].

Luby has constructed LT codes [12] that have degrees $O(\log n)$, are decoded in generator-representation form, and are provably capacity-achieving on the BEC under the following low-complexity algorithm:

Reversely analogous to [16], dualizing an LT code in generator form yields another code in parity check matrix form. Once in parity matrix form, we can transform this to a syndrome-former representation by adding dongles on checks to represent the coset constraints for the message index [17, sec. VIII.B]. The dual algorithm is as follows:

Proposition 5.1: Consider a linear code with generator matrix G and its dual code with $G^\perp = H$. The algorithm **ERASURE-DECODE-BEC**$(G, \underline{y})$ fails in step 5 if and only if the algorithm **ERASURE-ENCODE-DBC**$(H, \underline{s}, \underline{z})$ fails in step 5 where $\underline{y}$ has erasures specified by $\underline{e}$ and $\underline{z}$ has erasures specified by $\underline{e}^\perp = 1 - \underline{e}$.

The dual of an LT code [12] is successful with computational complexity of $O(n \log n)$. Figure 4 (L) gives an

ERASURE-ENCODE-DBC$(H, \underline{s}, \underline{z})$

1. **While** $\underline{z}$ has at least one erased sample **do**
2. **if** $\exists$ one z_i connected to exactly one neighbor check j **then**
3. Reserve z_i to later satisfy check j with syndrome s_j and erase check j.
4. **else**
5. **return** FAIL
6. **end if**
7. **end while**
8. Arbitrarily set unreserved erased z_i values.
9. Set reserved variables to satisfy the corresponding checks starting from the last reserved variable and working backward to the first reserved variable
10. **return** $\underline{z}$

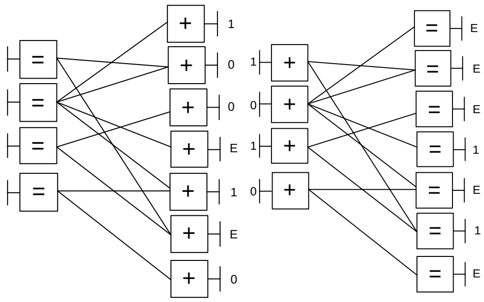

Fig. 4. (L) Generator Form LT Code for Decoding on a Binary Erasure Channel; (R) Syndrome Former Dual LT Code for Encoding on a Deterministic Broadcast Channel

example of decoding with a generator form LT code. The partially erased received sequence $\underline{y}$ lies on the right and the decoder must recover $\underline{u}$ corresponding to the non-existent symbols on the left. **ERASURE-DECODE-BEC**$(G, \underline{y})$ performs successfully here and the unique solution is given by $\underline{u} = (1, 1, 0, 0)$ and thus $\underline{u}G = (1, 0, 0, 1, 1, 1, 0)'$. (R) of Figure 4 gives the syndrome-former dual of the LT code in (L). Here, the syndrome is given on the left part of the graph by $\underline{s} = (1, 0, 1, 0)'$. The partially erased sequence $\underline{z}$ lies on the right and the encoder must recover $\underline{z}$. **ERASURE-ENCODE-DBC**$(H, \underline{s}, \underline{z})$ performs successfully here and one possible solution is given by $\underline{z} = (0, 0, 0, 1, 1, 1, 1)'$

VI. Capacity Achieving Codes for the Blackwell Channel

Our approach can also be used for the Blackwell channel, which is a classic example for the DBC model. In the Blackwell channel, the input is $X \in \{1, 2, 3\}$ with binary outputs. Its input-output relationship is as follows:

X	Y^1	Y^2
1	1	1
2	0	1
3	0	0

From here on out we will consider maximizing the sum-rate $R_1 + R_2$. Note that X and Y^1, Y^2 form a bijection and thus

$$H(X) = H\left(Y^1, Y^2\right). \tag{4}$$

Note that by (2), for a fixed $P(x)$ any set of achievable rates in $\mathcal{R}[P(X); f_1, f_2, \ldots, f_M]$ must satisfy $R_1 + R_2 \leq H\left(Y^1, Y^2\right)$. Since the uniform distribution uniquely maximizes entropy and since (4) holds, we can restrict our attention to rates $\mathcal{R}[P(X); f_1, f_2, \ldots, f_M]$ with $P(X)$ uniform. In this case,

$$\begin{aligned}\mathcal{R}[P(X); f_1, f_2, \ldots, f_M] = \big\{(R_1, R_2) \quad | \quad & R_1 \leq h_b(1/3), \\ & R_2 \leq h_b(1/3), \\ & R_1 + R_2 \leq \log_2 3\big\}\end{aligned}$$

where $h_b(\cdot)$ denotes the binary entropy function.

To operate at the corner point $(R_1, R_2) = (h_b(1/3), 2/3)$, we can first use enumerative source coding [11] to map message m_1 into a sequence $\underline{y}^1$ with $1/3$ ones and $2/3$ zeros as described in Section IV. Because of the structure of the Blackwell channel, whenever $Y_i^1 = 1$, we must have $Y_i^2 = 1$ as well. So communicating m_2 requires us to appropriately modulate the elements of $\underline{y}^2$ that are not constrained to be 1.

Specifically, to communicate the message m_2, we choose $\underline{y}^2$ to be a sequence consistent with $\underline{y}^1$ such that $H \cdot \underline{y}^2 = m_2$ where H is a fixed parity check matrix. Designing the parity check matrix, H, and constructing a low complexity encoding algorithm for $\underline{y}^2$ is then exactly the same problem as considered in Section V-A. By using duals of LT codes, it is possible to construct codes that allow us to select $\underline{y}^2$ so as to approach an information rate of $R_2 = 2/3$. The total rate at this corner point is $(h_b(1/3), 2/3)$ with a rate sum of

$$\begin{aligned}R_1 + R_2 &= \frac{1}{3}\log 3 + \frac{2}{3}\log\frac{3}{2} + \frac{2}{3} \\ &= \frac{1}{3}\log 3 + \frac{2}{3}\log 3 - \frac{2}{3}\log 2 + \frac{2}{3} = \log 3.\end{aligned}$$

The $(2/3, h_b(1/3))$ rate point can be achieved in a similar manner, and the remaining achievable rates can be attained via time-sharing.

References

[1] H. Weingarten, Y. Steinberg, and S. Shamai, “The capacity region of the Gaussian MIMO broadcast channel,” in *IEEE International Symposium on Information Theory*, Chicago, IL, June 2004.

[2] M. Costa, “Writing on dirty paper,” *IEEE Transactions on Information Theory*, vol. 29, no. 3, pp. 439–441, May 1983.

[3] U. Erez and S. ten Brink, “Approaching the dirty paper limit in canceling known interference,” in *Allerton Conference on Communication, Control, and Computing*, 2004.

[4] Y. Sun, A. D. Liveris, V. Stankovic, and Z. Xiong, “Near-capacity dirty-paper code design: A source-channel coding approach,” in *Conference on Information Sciences and Systems*, John Hopkins University, March 2005.

[5] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon limit error-correcting codes and decoding: Turbo codes,” *Proc. IEEE International Communications Conference*, 1993.

[6] R. Gallager, “Low-density parity-check codes,” *IRE Transactions on Information Theory*, vol. 8, pp. 21–28, Jan. 1962.

[7] K. Marton, “The capacity region of deterministic broadcast channels,” in *IEEE International Symposium on Information Theory*, Paris-Cachan, France, 1977.

[8] M. S. Pinsker, “Capacity of noiseless broadcast channels,” *Probl. Inform. Transm.*, pp. 97–102, 1978.

[9] K. Marton, “A coding theorem for the discrete memoryless broadcast channel,” *IEEE Transactions on Information Theory*, vol. 25, no. 3, pp. 306–311, 1979.

[10] T. P. Coleman, M. Effros, E. Martinian, and M. Medard, "Rate-splitting for the deterministic broadcast channel," in *IEEE International Symposium on Information Theory*, Adelaide, Australia, September 2005.
[11] T. Cover, "Enumerative source coding," *IEEE Transactions on Information Theory*, vol. 19, no. 1, pp. 73–77, Jan 1973.
[12] M. Luby, "LT codes," *Foundations of Computer Science*, pp. 271–280, November 2002.
[13] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, 1973.
[14] R. Gallager, *Information Theory and Reliable Communication*, John Wiley & Sons, New York, NY, 1968.
[15] I. Csiszár, "The method of types," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2205–2523, 1998.
[16] E. Martinian and J. S. Yedida, "Iterative quantization using codes on graphs," in *Allerton Conference on Communication, Control, and Computing*, 2003, http://arxiv.org/abs/cs.IT/0408008.
[17] G. D. Forney, "Codes on graphs: Normal realizations," *IEEE Transactions on Information Theory*, pp. 101–112, 2001.