Accepted for publication in the Proceedings of the 2008 IEEE International Symposium on Information Theory, to be held July 6-11, 2008, Toronto, Canada. ©IEEE 2008

Coding for the q-ary symmetric channel with moderate q

Claudio Weidmann
ftw. Telecommunications Research Center Vienna
Donau-City-Strasse 1, A-1220 Vienna, Austria
Email: claudio.weidmann@ieee.org

***Abstract*—We study coding schemes for the q-ary symmetric channel with moderate alphabet sizes q that are much smaller than the $q = 2^{256}$ considered as “entry level” in some recently proposed packet-based schemes. First, we show theoretical optimality of a simple layered scheme, then we propose a practical coding scheme based on a simple modification of standard binary LDPC decoding.**

I. Introduction

The q-ary symmetric channel (q-SC) with error probability p takes a q-ary symbol at its input and outputs either the unchanged input symbol, with probability $1-p$, or any of the other $q-1$ symbols, with probability $p/(q-1)$. It has attracted some attention recently as a more general channel model for packet-based error correction. For very large q, its normalized capacity approaches that of an erasure (packet loss) channel. In the following, we will only consider channel alphabets of size $q = 2^m$ with $m \in \mathbb{N}$.

The capacity of the q-SC with error probability p is

$$C_{q\text{-SC}} = m - h(p) - p\log(2^m - 1) \tag{1}$$

bits per channel use, where $h(p) = -p\log p - (1-p)\log(1-p)$ is the binary entropy function. (All logarithms in this paper are to the base 2.) Asymptotically in m, the normalized capacity $C_{q\text{-SC}}/m$ thus approaches $1-p$, which is the capacity of the binary erasure channel (BEC) with erasure probability p.

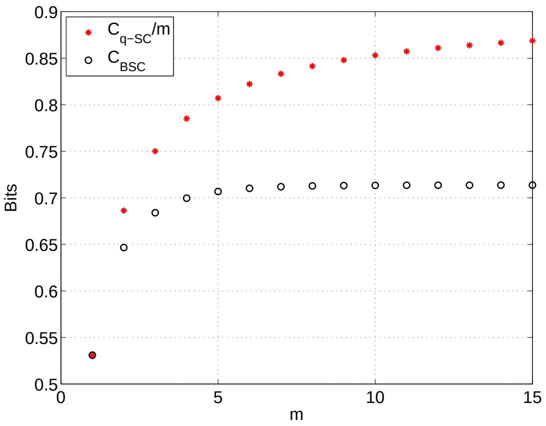

Fig. 1. Capacity of q-SC vs. marginal BSC (error probability $p = 0.1$).

Recent work [1], [2], [3], [4] has shown that it is possible to approach $C_{q\text{-SC}}$ for very large alphabet sizes $q = 2^m$, with symbols of hundreds to thousands of bits. The focus of the present work is on small q, with symbols of tens of bits at most, for which different coding techniques are needed.

The q-ary channel input and output symbols will be represented by binary vectors $\mathbf{x} = [x_1, x_2, \ldots, x_m]^T$ and $\mathbf{y} = [y_1, y_2, \ldots, y_m]^T$, respectively. A simplistic coding approach consists in decomposing the q-SC into m binary symmetric channels (BSC) with crossover probability

$$p_m = \frac{pq}{2(q-1)} = \frac{p}{2(1-2^{-m})}, \tag{2}$$

which have capacity $C_{\text{BSC}} = 1 - h(p_m)$ each.

We briefly study the normalized capacity loss $\Delta = C_{q\text{-SC}}/m - C_{\text{BSC}}$. For fixed m, we have $\lim_{p\to 0}\Delta = 0$; so using binary codes with independent decoders on the m-fold BSC decomposition might be good enough for small p (e.g. $p < 10^{-4}$). However, for p fixed, we have $\lim_{m\to\infty}\Delta = h(p/2) - p$, which can be a substantial fraction of the normalized q-SC capacity (e.g., for $p = 0.1$, $h(p/2) - p = 0.19$). Figure 1 shows that already for small m, the q-SC capacity is substantially larger than what can be achieved with the BSC decomposition. Clearly, there is a need for coding schemes for “large” p (say $p > 10^{-2}$), but moderate m (say $2 \le m < 20$). Figure 2 presents a different comparison, this time with fixed $m = 4$ and variable error probability p. The capacity gap in Figure 2 will widen with growing m, increasing the range of p for which specific q-SC coding is of interest .

II. Layered coding scheme

It is possible to build practical codes suited for iterative decoding that operate directly in the q-ary alphabet (e.g., LDPC codes over GF(q), [5]), but the decoder will in general suffer from an exponential increase (in m) of the size of the messages that need to be passed between stages. Thus even moderate m may be beyond reach of practical implementations with existing low-complexity decoding algorithms (e.g., [6]). As an alternative, we will study the following layered coding scheme based on binary codes. Blocks of k symbols $[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k]$ are split into m bit layers $[x_{i,1}, x_{i,2}, \ldots, x_{i,k}]$ and each layer is independently encoded with a code for a binary symmetric erasure channel (BSEC) with erasure probability δ_i and crossover probability ϵ_i, to be specified below. The key idea is to decode the layers in a fixed order and to declare *bit erasures* at those symbol positions in which a bit error occurred in a previously decoded layer. This saves on the code redundancy needed in the later layers, since erasures can be corrected with less redundancy than bit errors.

The decoder performs successive decoding of the m layers, starting from layer 1. All errors corrected at layer i and below are forwarded to layer $i+1$ as erasures, that is all bit error positions found in layers 1 up to i will be marked as erased in layer $i+1$, even though the channel provides a (possibly correct) binary output for those positions. Let ϵ_i be the probability that the channel output $\mathbf{y}$ is equal to the input $\mathbf{x}$ in positions 1 to $i-1$ and differs in position i (i.e., $[y_1, y_2, \dots, y_{i-1}] = [x_1, x_2, \dots, x_{i-1}]$ and $y_i \neq x_i$). A simple counting argument shows that there are 2^{m-i} such vectors $\mathbf{y}$ ($\neq \mathbf{x}$), out of a total $2^m - 1$. The i-th virtual subchannel is thus characterized by

$$\epsilon_i = \frac{2^{m-i}}{2^m - 1} p, \tag{3}$$

$$\delta_i = \sum_{j=1}^{i-1} \epsilon_j = \frac{2^m - 2^{m-i+1}}{2^m - 1} p. \tag{4}$$

Theorem 1: The layered scheme achieves q-SC capacity.

Proof: We assume an ideal scheme, in which all layers operate at their respective BSEC capacities and correct all errors and erasures. The BSEC(δ, ϵ) capacity is

$$C_{\text{BSEC}} = (1-\delta)\left(1 - h\left(\frac{\epsilon}{1-\delta}\right)\right). \tag{5}$$

Hence the sum of the layer rates becomes

$$\sum_{i=1}^{m} R_i = \sum_{i=1}^{m} (1-\delta_i)\left(1 - h\left(\frac{\epsilon_i}{1-\delta_i}\right)\right) \tag{6}$$

$$= m + \sum_{i=1}^{m} \Big\{ -\delta_i - (1-\delta_i)\log(1-\delta_i)$$

$$+ \epsilon_i \log \epsilon_i + (1-\delta_{i+1})\log(1-\delta_{i+1}) \Big\} \tag{7}$$

$$= m + \sum_{i=1}^{m} \big\{ -(m-i)\epsilon_i + \epsilon_i \log \epsilon_i \big\}$$

$$+ (1-p)\log(1-p) \tag{8}$$

$$= m + \sum_{i=1}^{m} \big\{ -(m-i)\epsilon_i + (m-i)\epsilon_i \big\} + p \log p$$

$$- p\log(2^m - 1) + (1-p)\log(1-p) \tag{9}$$

$$= m - h(p) - p\log(2^m - 1) = C_{q\text{-SC}},$$

where (6) follows from (5) and the definition of the layered scheme, (7) follows from $\delta_i + \epsilon_i = \delta_{i+1}$ (which holds up to $i = m$, when $\delta_{m+1} = p$), (8) follows from the evaluation of the telescoping sum and $\sum_{i=1}^{m} \delta_i = \sum_{i=1}^{m} (m-i)\epsilon_i$, and (9) follows from substituting (3) for ϵ_i. ■

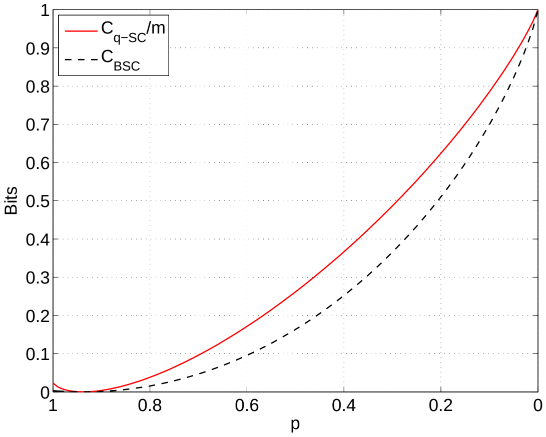

Fig. 2. Capacity of q-SC vs. marginal BSC (alphabet size $q = 2^4$).

The intuition behind the optimality of this (seemingly suboptimal) layered scheme is that once a bit error (and thus a symbol error) has been detected, all the following layers have bit error probability 1/2 in that position, since the q-SC assigns uniform probabilities over the possible symbol error values. Now the BSC(1/2) has zero capacity and so the concerned bits can be treated as erasures with no loss.

In view of the incremental capacity gains that are diminishing in m, as can be seen in Figure 1, an interesting variant of the layered scheme is to use $\mu < m$ BSEC layers as above, followed by a single "thicker" layer, which sends the remaining $m - \mu$ bits (per symbol) over the same BSEC($\delta_{\mu+1}, \epsilon_{\mu+1}$). In particular, this could even be beneficial in practical schemes, due to the trade-off between block size and capacity gap.

III. Symmetric coding scheme

The main disadvantage of the layered scheme is that in a practical implementation, each layer will need a different code that is tuned to the effective erasure and error probabilities resulting from the layers preceding it. This makes it impractical for hardware implementations, since the required silicon area would necessarily grow with the number of layers.

Ideally, a coding scheme for the q-SC should be symmetric in the bits composing a symbol, that is, no artificial hierarchy among bit layers should be introduced (notice that the order of the bit layers may be chosen arbitrarily). We propose to encode all bits composing the symbols with one "big" binary code, which needs to satisfy just slightly stricter constraints than an ordinary code for the BSC, while the decoder alone will exploit the knowledge about the underlying q-SC. The key concept from the layered scheme that should carry over is that the decoder is able to declare erasures at certain symbol positions and thus needs less error correction capability (for part of the bits in erased symbols).

Our proposal for a practical symmetric q-SC coding scheme is based on a binary low density parity check (LDPC) code with source block size $K = mk$ bits. We assume that the variable nodes in the LDPC decoder receive virtually independent extrinsic soft estimates of $X_{i,j}$ (that is, bit i of symbol/vector j, $j = 1, \dots, k$) from the check nodes. These amount to estimates of $P(X_{i,j}|\underline{\mathbf{Y}}_{[j]} = \underline{\mathbf{y}}_{[j]})$ or the corresponding log-likelihood ratio (LLR), $L(X) = \log(\Pr(X=1)/\Pr(X=0))$. Notice that this assumption does not actually hold, but it allows one to derive e.g. the standard binary variable-node decoding rule. As usual, the notation $\underline{\mathbf{y}}_{[j]}$ denotes the block consisting of all symbols/vectors $\mathbf{y}_t$ except the j-th (the underline reminds that this is a matrix for $m > 1$). This notation also makes clear what additional constraint the code should satisfy: we want an extrinsic estimate of $X_{i,j}$ that is independent of the other bits $X_{s \neq i,j}$ of the same symbol, so that in particular we may write $P(X_{i,j}|X_{1,j}, \ldots, X_{i-1,j}, \underline{\mathbf{Y}}_{[j]} = \underline{\mathbf{y}}_{[j]}) = P(X_{i,j}|\underline{\mathbf{Y}}_{[j]} = \underline{\mathbf{y}}_{[j]})$. A necessary condition for this is that the parity checks containing $X_{i,j}$ do not involve any of the bits $X_{s \neq i,j}$; this can be achieved by using an appropriate edge interleaver in the LDPC construction.

It turns out that the properties of the q-SC can be taken into account via a simple modification of the variable node computation in the message-passing decoding algorithm for binary LDPC codes. We factor the *a posteriori* probability (APP) of symbol $\mathbf{X}_j$ as follows:

$$P(\mathbf{X}_j|\underline{\mathbf{Y}} = \underline{\mathbf{y}}) \doteq P(\mathbf{Y}_j = \mathbf{y}_j|\mathbf{X}_j)P(\mathbf{X}_j|\underline{\mathbf{Y}}_{[j]} = \underline{\mathbf{y}}_{[j]}) \quad (10)$$

$$= P(\mathbf{Y}_j = \mathbf{y}_j|\mathbf{X}_j)\prod_{i=1}^{m} P(X_{i,j}|\underline{\mathbf{Y}}_{[j]} = \underline{\mathbf{y}}_{[j]}), \quad (11)$$

where the factorization in (11) is made possible by the above independence assumption (the symbol $\doteq$ denotes equality up to a positive normalization constant). Using the definition of the q-SC, this becomes

$$P(\mathbf{X}_j = \mathbf{x}_j|\underline{\mathbf{Y}} = \underline{\mathbf{y}}) \doteq \begin{cases} (1-p)\prod_{i=1}^{m} P(X_{i,j} = x_{i,j}|\underline{\mathbf{Y}}_{[j]} = \underline{\mathbf{y}}_{[j]}), & \mathbf{x}_j = \mathbf{y}_j, \\ \frac{p}{2^m-1}\prod_{i=1}^{m} P(X_{i,j} = x_{i,j}|\underline{\mathbf{Y}}_{[j]} = \underline{\mathbf{y}}_{[j]}), & \mathbf{x}_j \neq \mathbf{y}_j. \end{cases} \quad (12)$$

We define the extrinsic probability that $\mathbf{X}_j = \mathbf{y}_j$ as

$$\beta_j = \prod_{i=1}^{m} P(X_{i,j} = y_{i,j}|\underline{\mathbf{Y}}_{[j]} = \underline{\mathbf{y}}_{[j]}) = \prod_{i=1}^{m} p_{i,j}, \quad (13)$$

where we introduced $p_{i,j} = P(X_{i,j} = y_{i,j}|\underline{\mathbf{Y}}_{[j]} = \underline{\mathbf{y}}_{[j]})$ for convenience. The normalization constant in (12) thus becomes

$$\gamma_j = (1-p)\beta_j + \frac{p}{2^m-1}(1-\beta_j). \quad (14)$$

Then the bit APP may be be obtained by the marginalization

$$P(X_{i,j} = x_{i,j}|\underline{\mathbf{Y}} = \underline{\mathbf{y}}) = \sum_{\mathbf{x}_{[i],j} \in \{0,1\}^{m-1}} P(\mathbf{X}_j = \mathbf{x}_j|\underline{\mathbf{Y}} = \underline{\mathbf{y}}), \quad (15)$$

which may be written as

$$P(X_{i,j} = x_{i,j}|\underline{\mathbf{Y}} = \underline{\mathbf{y}}) = \begin{cases} \left[(1-p)\beta_{[i],j} + \frac{p}{2^m-1}(1-\beta_{[i],j})\right] \cdot \frac{p_{i,j}}{\gamma}, & x_{i,j} = y_{i,j}, \\ \frac{p}{2^m-1}P(X_{i,j} = x_{i,j}|\underline{\mathbf{Y}}_{[j]} = \underline{\mathbf{y}}_{[j]})\gamma^{-1}, & x_{i,j} \neq y_{i,j}, \end{cases} \quad (16)$$

where $\beta_{[i],j} = \beta_j / p_{i,j}$ is the "intra-symbol" extrinsic probability that $\mathbf{X}_j = \mathbf{y}_j$, using no information on bit $X_{i,j}$. Finally, we may express the *a posteriori* bit-level LLR as

$$\begin{aligned} &L_{\text{app}}(X_{i,j} = y_{i,j}) \\ &= \log\left(\frac{(2^m-1)\beta_j - 2^m p\beta_j + pp_{i,j}}{p(1-p_{i,j})}\right) \\ &= \log\left(\frac{(2^m-1)\beta_{[i],j} - 2^m p\beta_{[i],j} + p}{p} \cdot \frac{p_{i,j}}{1-p_{i,j}}\right) \\ &= \log\left(1 + \frac{2^m - 2^m p - 1}{p} \cdot \beta_{[i],j}\right) + L_{\text{extr}}(X_{i,j} = y_{i,j}). \end{aligned} \quad (17)$$

(The usual $L(X)$ is obtained from $L(X=y) = \log(\Pr(X=y)/\Pr(X=1-y))$ via a sign flip depending on y.)

The second term in (17) corresponds to the extrinsic information from the check nodes that is processed at the variable nodes in order to compute the bit APP in standard binary LDPC decoding. The difference lies in the first term in (17), which is the equivalent of the channel LLR, $L_{\text{ch}}(X) = \log(P(y|X=1)/P(y|X=0))$, in the standard binary case. When the extrinsic information on the bits $X_{s \neq i,j}$ favors the hypothesis $\mathbf{X}_j \neq \mathbf{y}_j$, the product $\beta_{[i],j}$ will be small and therefore the channel LLR in (17) will be close to zero, which is equivalent to declaring a bit erasure. This shows that the symmetric LDPC scheme relies on "distributed" *soft* bit erasure estimates, while in the layered scheme the erasures are declared in a *hard* "top-down" fashion.

Equation (17) describes the main modification required to turn a binary message passing LDPC decoder into one for the q-ary symmetric channel. For practical implementation purposes, it should probably be modified (approximated) in order to avoid switching back and forth between probabilities and LLRs when computing β_j. A final detail is the specification of the initial channel LLR in (15), which is needed to start the decoder iterations. By inserting the memoryless worst-case estimate $\beta_{[i],j} = 2^{-m+1}$ into (15), we obtain the channel term

$$\log\left(1 + \frac{2^m - 2^m p - 1}{p} \cdot \beta_{[i],j}\right) = \log\left(\frac{2(1-2^{-m}) - p}{p}\right), \quad (18)$$

which is exactly the channel LLR for the marginal BSC with crossover probability p_m given in (2).

Notice that the decoder iterations are still exclusively between variable and check node computations, like in the binary case. However, computing the bit-level L_{app} at the variable nodes requires the extrinsic information (the check node messages) for all bits within a symbol; this might be considered an additional level of message exchanges (specifically, plain copying of messages), but it does not involve iterations of any kind. The complexity increase compared to binary LDPC decoding is on the order of at most m operations per variable node, depending on the scheduling. In fact, the marginalization operation (15) is reminiscent of a combined detector and variable-node decoder for binary LDPC codes that are directly mapped to larger (MIMO) signal constellations [7]. Thanks to the symmetry of the q-SC, here it is not necessary to

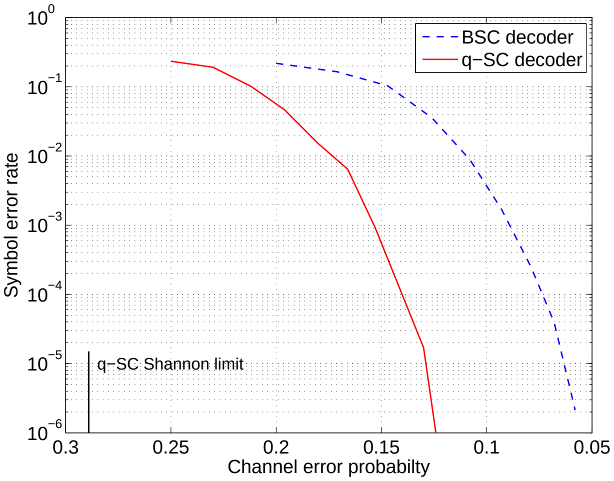

Fig. 3. q-SC vs. BSC rate 1/2 LDPC decoding ($q = 2^4$, $k = 150$ symbols).

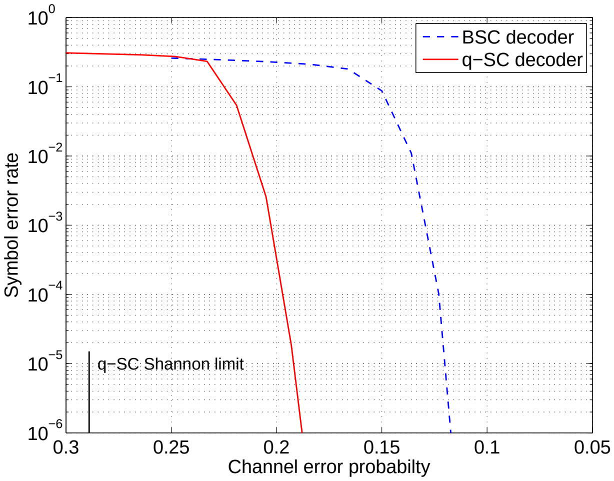

Fig. 4. q-SC vs. BSC rate 1/2 LDPC decoding ($q = 2^4$, $k = 1500$ symbols).

actually sum over all 2^m symbol values. Other similar work includes iterative demapping and decoding [8], [9], which however involves proper iterations between the demapper and the LDPC decoder, that are treated as two separate functional blocks.

IV. Simulation results

To assess the performance gains of an LDPC decoder adapted to the q-SC compared to an ordinary binary decoder (for the BSC decomposition outlined in Sec. I), we ran simulations using a standard $(3, 6)$-regular, rate 1/2 LDPC code (see, e.g., [10, Ch. 3]). No particular interleaving measures were taken that would have guaranteed the (quasi-)independence of the extrinsic bit estimates within a symbol. Thus plenty of optimization opportunities are left open, be it in the code (irregular LDPC) or the interleaver. Exactly the same encoder and channel model are used, once with the standard BSC decoder, and once with the q-SC decoder. Results for $m = 4$ and block length $k = 150$ symbols ($N = 1200$ channel bits) are shown in Fig. 3; results for block length $k = 1500$ ($N = 12000$) are shown in Fig. 4. Although performance is far from the Shannon limit (because the code is not optimized), the advantage of the q-SC decoder is clearly visible.

V. Outlook

We expect to have first results on practical code optimization in time for the conference.

Acknowledgments

The author would like to thank Gottfried Lechner for kindly providing his LDPC toolbox and the $(3, 6)$-regular code used in the simulations, as well as for helpful discussions. Special thanks go to Jossy Sayir, who pointed out the similarity with iterative demapping and decoding and whose in-depth critique helped improve the presentation of the paper.

The Telecommunications Research Center Vienna (ftw.) is supported by the Austrian Government and the City of Vienna within the competence center program COMET.

References

[1] M. G. Luby and M. Mitzenmacher, “Verification-based decoding for packet-based low-density parity-check codes,” *IEEE Trans. Inform. Theory*, vol. 51, no. 1, pp. 120–127, January 2005.

[2] A. Brown, L. Minder, and A. Shokrollahi, “Probabilistic decoding of interleaved RS-codes on the q-ary symmetric channel,” in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Chicago, USA, June 27 – July 2, 2004.

[3] A. Shokrollahi and W. Wang, “Low-density parity-check codes with rates very close to the capacity of the q-ary symmetric channel for large q,” in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Chicago, USA, June 27 – July 2, 2004.

[4] A. Shokrollahi, “Capacity-approaching codes on the q-ary symmetric channel for large q,” in *Proc. ITW*, San Antonio, Texas, USA, October 24 – 29, 2004.

[5] M. C. Davey and D. MacKay, “Low density parity check codes over GF(q),” *IEEE Commun. Lett.*, vol. 2, pp. 165–167, June 1998.

[6] D. Declercq and M. Fossorier, “Decoding algorithms for nonbinary LDPC codes over GF(q),” *IEEE Trans. Commun.*, vol. 55, no. 4, pp. 633–643, April 2007.

[7] S. ten Brink, G. Kramer, and A. Ashikhmin, “Design of low-density parity-check codes for modulation and detection,” *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 670–678, April 2004.

[8] A. Sanderovich, M. Peleg, and S. Shamai, “LDPC coded MIMO multiple access with iterative joint decoding,” *IEEE Trans. Inform. Theory*, vol. 51, no. 4, pp. 1437–1450, April 2005.

[9] G. Lechner, J. Sayir, and I. Land, “Optimization of LDPC codes for receiver frontends,” in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Seattle WA, USA, July 9–14, 2006.

[10] R. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, to appear 2007, see also http://lthcwww.epfl.ch/mct/index.php.