Chapter 16

COOPERATIVE COMMUNICATION WITH MULTIPLE DESCRIPTION CODING

Expanding the Cooperation to the Realm of Source Encoding

Morten Holm Larsen
Aalborg University - Department of Telecommunication Technology
mhl@kom.aau.dk

Petar Popovski
Aalborg University - Department of Telecommunication Technology
petarp@kom.aau.dk

Søren Vang Andersen
Aalborg University - Department of Telecommunication Technology
sva@kom.aau.dk

Abstract: Multiple Description Coding (MDC) is a source coding technique where the source is encoded into two or more descriptions. The descriptions are self–sufficient in the sense that each description can provide a distorted version of the source information, while the distortion is decreased as more descriptions are utilized at the decoder. MDC was proposed as a coding scheme to gain robustness to packet loss over a communication network in a scenario with single source and single destination. In this chapter, we study how the MDC can be applied to support cooperative communications. In particular, we focus on Multiple Description Lattice Vector Quantizer (MDLVQ) and suggest optimal design methods for MDLVQ in a cooperative network. Next, we propose a novel scheme, termed MDC with Conditional Compression (MDC–CC). The basic observation behind MDC–CC is that the availability of timely feedback presents the need for robustness, originally implied by the MDC scheme. The source encoding with

Frank H. P. Fitzek and Marcos D. Katz (eds.), Cooperation in Wireless Networks: Principles and Applications, 515–545.

MDC–CC is done in such a way that upon having a feedback from the destination, the encoding overhead can be removed at any time by a cooperative node or an active network. We show an implementation where MDC–CC utilizes the highly structured design of MDLVQ and thus produces a very elegant solution, where the computational complexity becomes almost negligible. Finally, we introduce three generic scenarios for cooperative communication with MDC and, in particular, MDC–CC: data delivery with cooperative sources, data delivery with cooperative destinations and data delivery with meshed cooperation.

Keywords: higher layer cooperative networking, source coding, lattice vector quantizers, multiple description coding (MDC), MDC with conditional compression (MDC–CC), cooperative communication, cooperative sources, cooperative destinations, meshed cooperation.

1. Introduction

The paradigm of cooperative communication has recently gained significant attention in relation to the wireless communications. The initiating observation is that the broadcast nature of the wireless medium offers a possibility for a group of terminals to cooperate by sharing their antennas and thus creating a distributed multi–antenna entity, see *e.g.* [Nosratnia et al., 2004]. In an exemplifying scenario, such entity communicates with the Base Station (BS), this provides the terminals with a better service as compared to the case when each terminal communicates with the BS independently. Such method of communication creates a diversity effect, termed *cooperative diversity* and is concerned mainly with the physical and link layer of the protocol stack. At the network layer cooperation also occurs as explained *e.g.* by [Gupta and Kumar, 2000]. As an example, in multi–hop wireless networks, a communication node can act as a router that forwards packets on behalf of other nodes.

The instances of cooperative communication mentioned above are exploiting the benefits of cooperative communication, regardless of the information source that produces the communicated data. The solution space for cooperative communication can be further expanded by bringing the cooperation further up in the protocol stack and considering the source coding aspect. In this chapter we propose and analyze the suitability of source coding schemes based on Multiple Description Coding (MDC) for the scenarios with cooperative communication. We give examples of how to design the source coding schemes within the multiple description (MD) paradigm to account for the fact that the descriptions are transmitted through a cooperative communication network.

Source Coding and Cooperative Communication

To get started with the cooperative aspects of the source encoding, in this section we introduce several motivating examples. First, let us consider an example of cooperative networking where two terminals cooperate on a realtime application download, such as a video–on–demand (VoD). A straightforward cooperation scheme can be one in which the source sends the whole information to one of the terminals and that terminal forwards the information to the other terminal. With a more sophisticated scheme, a coarse information about the source is sent to both terminals and a different refinement of information is sent to each terminal. If a terminal receives only the information sent directly from the source to it, then the video is shown with a low quality. To obtain the full video quality, the terminal should receive also the refinement information sent to the other terminal. In this simple example, the cooperation among the terminals, *i.e.* the exchange of refinement information, increases the video quality.

The theoretical framework for splitting information is known in the information theory as Multiple Description (MD) coding. MD was first introduced in the 1970s. A thorough review of the history of MD with applications and algorithms can be found in [Goyal, 2001]. The encoding of the source information is done in a way that multiple descriptions are produced to describe each chunk of source information. Each description contains coarse information about the source, but by using all generated descriptions this can be completely recovered. The fundamental MD encoding–decoding concept is typically described and analyzed in a framework with two channels and three receivers, as shown in Figure 16.1.

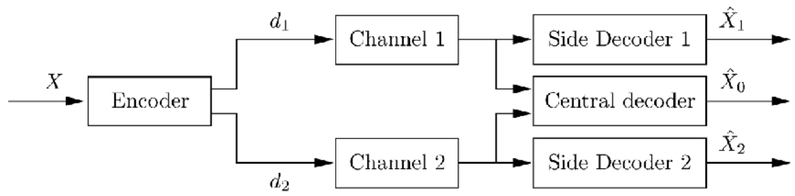

Figure 16.1. Source encoder (at the information source node) and source decoder (at the destination node).

In Figure 16.1 the encoder sends information about the source over two channels. When only one description reaches the destination, the destination node applies the side decoder and reconstructs the source with a low quality. If both descriptions reach the destination, then the central decoder is used and source information is reconstructed with a high quality. The system on Figure 16.1 can be generalized to M channels and $2^M - 1$ receivers.

Now, Figure 16.2 depicts a simple cooperative network, where BS is transmitting the two descriptions d_1 and d_2 to terminal 1 and terminal 2, respectively. Subsequently, the two terminals exchange descriptions through the cooperative link.

In this system, if terminal 1 receives only d_1 from BS, it reconstructs the source with a low quality. If, in addition to d_1, terminal 1 receives d_2 from

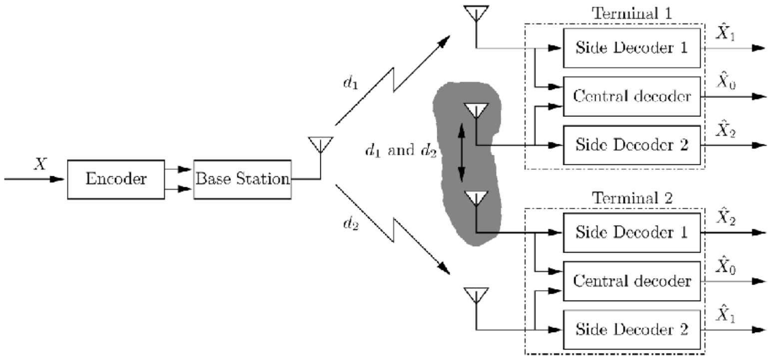

Figure 16.2. A simple cooperative network with two terminals and a Base Station (BS). The gray cloud denotes the cooperative link.

terminal 2, it applies the central decoder and hence reconstructs the source with a high quality. In the case where d_1 does not reach terminal 1, but only d_2 is received through the cooperative link, the second side decoder is used and the source is reconstructed with a low quality. This mechanism constitutes a first example on how Multiple Descriptions can be used in Cooperative Communication.

The incorporated robustness in the MD introduces a rate overhead for similar distortion level when compared with encoding the source for a single channel and decoder. Hence, when designing the encoder and decoder we can choose the amount of overhead. Conversely, this means that when the quality provided by the central decoder is fixed, the higher overhead we allow, the higher quality is provided by each side decoder. In the classical MD framework shown on Figure 16.1, the overhead and the design of encoder and decoders are determined from the rates and loss probabilities on the two channels by [Østergaard et al., 2004]. In practice, the loss probabilities are normally not known explicitly, but only estimates of the loss probabilities are known with an uncertainty. In [Larsen et al., 2005] it is shown how this uncertainty can be taken into account when designing the encoder and decoders.

Broadcast is frequently used in streaming real–time data to many users in a wireless environment. The existing broadcast schemes use a special case of MD commonly referred to as *layered coding*. In layered coding, the first description contains a coarse information and the following descriptions are only containing refinement information. Thus, in the case where the first description is lost, the following descriptions are approximately useless. Under a certain assumption, the combination of cooperative networking and multiple descriptions is also advantageously applied in the broadcast scenario. The critical assumption is the existence of a fast feedback between the cooperating terminals. If the terminals cooperate with a fast feedback mechanism, then in case of information loss over the broadcast channel, a terminal can request the information from the other terminals via the fast feedback. An exemplifying scheme for this method is shown in Figure 16.3.

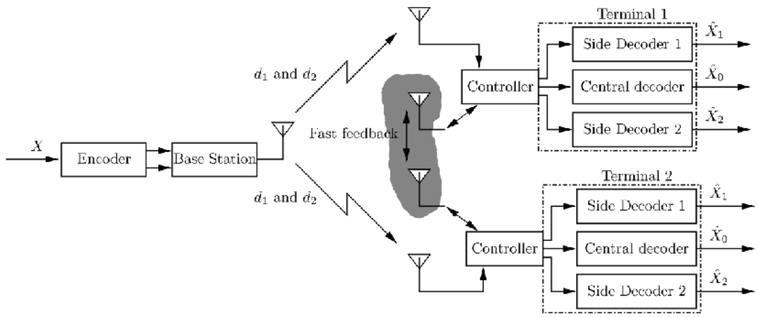

Figure 16.3. Scenario for cooperative reception of broadcast information with base station and two terminals.

Organization of this Chapter

After briefly introducing the motivating examples above, the next section provides sufficient mathematical detail on Multiple Description Coding (MDC) to do the analysis of its use for Cooperative Networking. In particular, we focus on the Multiple Description Lattice Vector Quantizer (MDLVQ) as we will use the lattice structure in the design examples throughout this chapter. Subsequently, Section 3 describes how to optimize the MD quantizer design for the cooperative scenario from Figure 16.2, and Section 3 further describes how the design result from [Østergaard et al., 2004] and [Larsen et al., 2005] can be used to design MD quantizers for the cooperative scenario from Figure 16.2. In Section 4 we introduce a novel MDC scheme, termed *MDC with Conditional Compression (MDC–CC)*. With MDC–CC, the compression of the source information can be done by any node in the network after such node gets information about what has already been received at the destination. We show that by using a lattice vector quantizer, the MDC–CC scheme becomes very elegant and the computational complexity becomes almost negligible. Section 5 discusses the presented methods and casts them into a set of more generally defined scenarios in which the combination of MDC and cooperative networking should be considered. Finally, the last section concludes the chapter.

2. Multiple Description Coding (MDC) Basics

In an MD system for two channels, the encoder sends information about the source over the two channels, with rate R_i bits per source symbol [bpss] for each channel, $i \in \{1, 2\}$. Each channel may either be in working or non–working state and this is not known at the encoder. The destination node uses side decoder 1 to reconstruct the source when channel 1 is in the working state and channel 2 is in the non–working state. Similarly, side decoder 2 is used to reconstruct the source when channel 2 is in the working state and channel 1 is in the non–working state. When both channels are in the working state the central decoder is used to reconstruct the source. Although this system can be generalized to M channels and $2^M - 1$ receivers, all fundamental concepts presented in this chapter can be explained in the simpler two channel case.

The principles in this chapter applies with multiple description designs in general, such as the scalar and vector designs proposed in [Vaishampayan, 1993; Vaishampayan and Domaszewicz, 1994]. However, in the special case of lattice structured quantizers, the principles have elegant implementations with very low computational complexity. The design framework of lattice structured multiple description quantizers is extensively described and analyzed in the literature. See *e.g.* [Sergio et al., 1999; Vaishampayan et al., 2001; Goyal et al., 2002] for a thorough introduction to this field. In the following, we give an outline of the main issues from this framework that we will need to convey central concepts related to design for cooperative networks.

Lattice Vector Quantizer

Let the real lattice $\Lambda \subset \mathbb{R}^L$ be a collection of lattice points λ (reconstruction points), where Λ is generated by a generating matrix $\mathbf{G} \in \mathbb{R}^{L\times L}$:

$$\Lambda = \{\lambda : \lambda = \mathbf{G}\xi, \xi \in \mathbb{Z}^L\}. \tag{16.1}$$

The generating matrix $\mathbf{G}$ is a set of linearly independent basis vectors $\mathbf{v}$, which span the lattice, $\mathbf{G} = [\mathbf{v}_1\ \mathbf{v}_2\ \cdots\ \mathbf{v}_L]$. The region of source vectors $\mathbf{x}$ that quantize to a lattice point λ is called a Voronoi region $V(\lambda)$, and given by

$$V(\lambda) \triangleq \{\mathbf{x} \in \mathbb{R}^L : \|\mathbf{x} - \lambda\|^2 \leq \|\mathbf{x} - \lambda^*\|^2, \forall\lambda^* \in \Lambda\}, \tag{16.2}$$

where the distortion measure $\|\cdot\|^2 \triangleq \frac{1}{L}\mathbf{x}^T\mathbf{x}$ is chosen to be the normalized 2–norm. The normalized 2–norm is typically used, even in the cases for which the map between 2–norm and the perceptual distortion measure is complex. In such cases, *e.g.* voice, audio and video coding, such map is typically provided by use of compandors and weighting filters. The L dimensional volume of a Voronoi region is the determinant of the generator matrix

$$\nu = \det[G]. \tag{16.3}$$

Figure 16.4 shows two very common lattices in two-dimensional space, Z_2 and A_2, where the generating matrices are

$$Z_2 : \mathbf{G} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ and } A_2 : \mathbf{G} = \begin{bmatrix} 1 & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} \end{bmatrix}. \tag{16.4}$$

The encoding algorithm is straightforward for Z_L lattice, but for A_L the encoding algorithm is nontrivial. A fast encoding algorithm for A_L is given in [Conway and Sloane, 1982a].

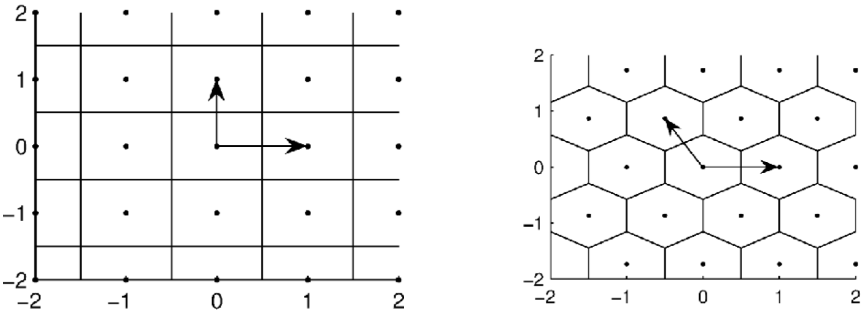

Figure 16.4. An example of a Z_2 lattice (left) and an A_2 lattice (right). The dots are lattice points λ, the lines bounds the voronoi regions and arrows are the basis vectors.

When evaluating the average distortion for a given lattice quantizer, let $X \in \mathbb{R}^L$ be an arbitrary i.i.d source and $f_{\mathbf{X}}(\mathbf{x})$ be the probability density function (pdf).

$$D = \sum_{\lambda \in \mathbf{\Lambda}} \int_{V(\lambda)} f_{\mathbf{X}}(\mathbf{x}) \|\mathbf{x} - \lambda\|^2 d\mathbf{x}. \tag{16.5}$$

Often, it is useful to analyze the distortion for high resolution of the quantizer. In high resolution, the Voronoi region is small and we can assume a locally constant probability density, $f_{\mathbf{X}}(\mathbf{x}) \approx f_\lambda$ for $\mathbf{x} \in V(\lambda)$. Thus, we can find the probability for a given λ by

$$P_\lambda \triangleq Pr\left(X \in V(\lambda)\right) \approx f_\lambda \nu_\lambda, \tag{16.6}$$

where ν_λ denotes the volume of the Voronoi region, $V(\lambda)$. From the structure of the lattice, we see that the volume is constant and Eq. (16.5) can be written as

$$D \approx \sum_{\lambda \in \mathbf{\Lambda}} \frac{P_\lambda}{\nu} \int_{V(\lambda)} \|\mathbf{x} - \lambda\|^2 d\mathbf{x}. \tag{16.7}$$

Quantization error will, due to the geometrical structure, yield

$$\int_{V(\lambda)} \|\mathbf{x} - \lambda\|^2 d\mathbf{x} = \int_{V(0)} \|\mathbf{x}\|^2 d\mathbf{x}, \tag{16.8}$$

where $V(0) = V(\lambda_{\mathbf{0}})$ and $\lambda_{\mathbf{0}} = [00 \cdots 0]^T$, [Gray, 1990]. Traditionally, the normalized second–order moment $G(\mathbf{\Lambda})$ is defined as

$$G(\mathbf{\Lambda}) \triangleq \frac{\int_{V(0)} \|\mathbf{x}\|^2 d\mathbf{x}}{\nu^{1+2/L}}, \tag{16.9}$$

and has been tabulated for many lattice structures, see *e.g* [Conway and Sloane, 1999; Conway and Sloane, 1982b]. Few of these are summarized in Table 16.1. The average distortion can hence be found from the volume of the Voronoi region and this table as

$$D \approx G(\mathbf{\Lambda})\nu^{2/L}. \tag{16.10}$$

For an arbitrary volume, we can observe from Eq. (16.9) that the smallest second–order moment is obtained by a sphere lattice for the 2–norm. Spheres can not be packed to fill the space and therefore cannot be used to constitute a quantizer, but the second moment of a sphere gives an analytical lower bound. However, from the table we see that A_2 is very close to the second moment of the sphere, and it is well known that A_2 is the optimal lattice for two dimensions, see *e.g.* [Conway and Sloane, 1999]. Unfortunately, the problem of constructing an optimal lattice for higher dimensions is still unsolved, but there exist lattices that perform relatively close to the sphere lower bound in higher dimensions, *e.g.* the Leech lattice in 24 dimensions, Λ_{24}. When analyzing the lattice vector

$\mathbf{\Lambda}$	$G(\mathbf{\Lambda})$	$n \to \infty$	$n = 24$	$n = 2$
Z_n	$\frac{1}{12}$	$\frac{1}{12}$	-	$\frac{1}{12}$
A_n	$\frac{1}{(n+1)^{1/n}}\left(\frac{1}{12}+\frac{1}{6(n+1)}\right)$	$\frac{1}{12}$	-	0.0802
Λ_{24}	Monte Carlo Simulation	-	0.06561	-
Sphere	$\frac{\Gamma(n/2+1)^{2/n}}{(n+2)\pi}$	0.0585	0.0647	0.0796

Table 16.1. Second moment for the most popolar lattice, ([Conway and Sloane, 1999]).

quantizer, we assume an entropy coding that maps a source symbol ξ to a variable bit rate. The mapping is made such that the length of the binary sequence to which ξ is mapped is inversely proportional to the probability of occurrence of ξ. This mapping is exploited in several data compression algorithms such as Huffman coding, see *e.g* [Cover and Thomas, 1991]. It can be shown that a Huffman coder can encode to an average bit rate of the entropy plus 1 bit. From this point we will assume that the entropy coder can encode arbitrarily close to the entropy. The entropy in bit per dimension for a lattice vector quantizer, when assuming high resolution is given by [Gray, 1990]:

$$R = -\frac{1}{L}\sum_{\lambda\in\mathbf{\Lambda}}\int_{V(\lambda)} f_{\mathbf{x}}(\mathbf{x})d\mathbf{x}\log_2\int_{V(\lambda)} f_{\mathbf{x}}(\mathbf{x})d\mathbf{x} \tag{16.11}$$

$$\approx h(X) - \frac{1}{L}\log_2(\nu). \tag{16.12}$$

To summarize, we list our observations about the benefits that emerge from the highly structured nature of the lattice:

- As any lattice point λ can be regenerated from ξ and the generating matrix $\mathbf{G}$, there is no need to store the reconstruction point.

- A fast encoding algorithm exists for Z_n and A_n. This is described in [Conway and Sloane, 1982a].
- There is a closed–form expression for the entropy, and hence an expression of the achievable average bit rate.
- There is a simple expression for the average distortion for the lattice quantizer. The second moment of this distortion is typically given in the form of a table.

We'll make beneficent use of each of these properties when we return to cooperative communication later in this chapter.

EXAMPLE 16.1 *Let us design a two dimensional lattice vector quantizer for a system with a unit variance Gaussian source, a A_2 lattice and with a rate constraints $R = 3$ bit pr. dimension. The differential entropy of a unit variance Gaussian source is $h(X) = \frac{1}{2}\log_2(2\pi e)$, [Cover and Thomas, 1991]. First we find the volume of the Voronoi region by Eq. (16.11),*

$$\nu = 2^{L(h(X)-R)} = 0.1334. \tag{16.13}$$

Next, we determine the generator matrix for this system by scaling the matrix for A_2 given by Eq. (16.4) as,

$$\mathbf{G}' = \mathbf{G}c. \tag{16.14}$$

The scaling constant "c" can be determined from the volume constrain in Eq. (16.13),

$$\begin{aligned} \nu &= \mid \mathbf{G}c \mid & (16.15)\\ &= \prod_i^L \lambda_i c & (16.16)\\ &= c^L \mid \mathbf{G} \mid, & (16.17) \end{aligned}$$

where λ_i is the i'th eigenvector of $\mathbf{G}$. The generator matrix can now be determined as,

$$\mathbf{G}' = \mathbf{G}\sqrt[L]{\frac{\nu}{\mid \mathbf{G} \mid}} = \begin{bmatrix} 0.3925 & -0.1963 \\ 0 & 0.3399 \end{bmatrix}. \tag{16.18}$$

Now, for example, the source input $x = [-0.1\ 0.6]^T$ is quantized to $\lambda = [0\ 0.6799]^T$ or $\xi = [1\ 2]^T$. This can be realized from Figure 16.4 by dividing the source input x by c.

Review of the Geometrical Relationship

Before explaining the method for constructing the unbalanced multiple description vector quantizer (MDLVQ) proposed in [Diggavi et al., 2002], we now review some important geometrical relationships between two lattices. We define a sublattice $\mathbf{\Lambda}'$ to be geometrically similar to the lattice $\mathbf{\Lambda}$, if $\mathbf{\Lambda}' \subseteq \mathbf{\Lambda}$ and $\mathbf{\Lambda}'$ can be obtained by scaling and rotating $\mathbf{\Lambda}$. The generator matrix for the $\mathbf{\Lambda}'$ is given by

$$\mathbf{\Lambda}' = \mathbf{UG}. \tag{16.19}$$

The number of lattice points λ that are included in a Voronoi region of the sublattice $V'(\lambda')$ is denoted N. The requirements for similar sublattices of Z_2 and A_2 are derived in [Conway et al., 1999], where it is found that for Z_2 there exists a similar sublattice when N has the form $N = a^2 + b^2$, where $a, b \in \mathbb{Z}$. The possible combinations of a and b yields

$$N = 1, 2, 4, 5, 8, 9, 10, 13, 16, 17, 18, 20, \cdots \tag{16.20}$$

This result can be found in [Sloane, 2005] as sequence A1481. The rotating and scaling matrix $\mathbf{U}$ for the similar sublattice of Z_2 is found by

$$\mathbf{U} = \begin{bmatrix} a & -b \\ b & a \end{bmatrix}. \tag{16.21}$$

For A_2 there exists a similar sublattice when N has the form $N = a^2 - ab + b^2$, where $a, b \in \mathbb{Z}$. The possible combinations of a and b yields the sequence,

$$N = 1, 3, 4, 7, 9, 12, 13, 16, 19, 21, 25, 27, \cdots \tag{16.22}$$

which again can be found in [Sloane, 2005], sequence A3136. The rotation and scaling matrix for A_2 is

$$\mathbf{U} = \begin{bmatrix} a + b\cos(2\pi/3) & -b\sin(2\pi/3) \\ b\sin(2\pi/3) & a + b\cos(2\pi/3) \end{bmatrix}. \tag{16.23}$$

Figure 16.5 shows an example of a geometrically similar sublattice of A_2, where $a = 4$ and $b = 3$. It can be seen from the figure that the sublattice includes exactly $N = 13$ lattice points, as expected. We define a sublattice $\mathbf{\Lambda}'$ to be clean if $\mathbf{\Lambda}$ does not intersect with the boundary of the Voronoi region of $\mathbf{\Lambda}'$. The geometrical relationship for a clean respectively non–clean sublattice of Z can be illustrated as in Figure 16.6, where $\mathbf{\Lambda}$ and two sublattices $\mathbf{\Lambda}'$ are generated by an even and an odd scaling, respectively. From this figure we observe how an even N will result in intersection between lattice points and the boundary of the $\mathbf{\Lambda}'$ Voronoi region. Conversely, when N is odd, the boundary will not intersect with lattice points and hence the sublattice is clean.

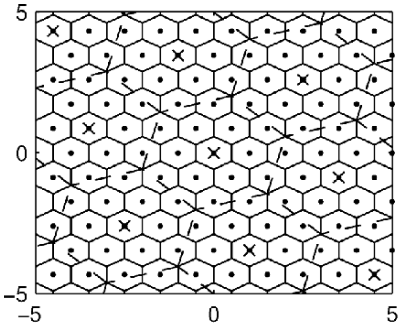

Figure 16.5. An example of an A_2 lattice and a similar sublattice (dashed) with $N = 13$.

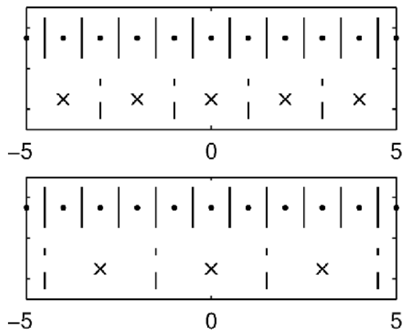

Figure 16.6. In the top an example of a Z_1 lattice and a clean sublattice where $N = 3$. In the bottom a sublattice where $N = 2$ and not clean.

The condition for a clean sublattice in two dimensions is investigated in [Conway et al., 1999] and clean sublattice of Z_n for higher dimensions are solved in [Diggavi et al., 2002]. Condition for a clean sublattice of Z_2 is that N must be odd and have the form $N = a^2 + b^2$, where $a, b \in \mathbb{Z}$. The possible combinations of a and b yields

$$N = 1, 5, 9, 13, 17, 25, 29, 37, 41, \cdots \tag{16.24}$$

as can be found in [Sloane, 2005], sequence A57653. For making a clean sublattice of A_2, N must have the form $N = a^2 - ab + b^2$, where a and b are relatively prime. This condition yields the possible combinations,

$$N = 1, 7, 13, 19, 31, 37, 43, 49, 61, \cdots \tag{16.25}$$

as can be found in [Sloane, 2005], sequence A57654. The rotating and scaling matrix $\mathbf{U}$ for clean sublattice of Z_2 and A_2 are expressed in Eq. (16.21) and Eq. (16.23), respectively. In the example on Figure 16.5 with $N = 13$, it can be seen that the sublattice is both similar and clean.

Multiple Description Lattice Vector Quantizers

The task of designing a Multiple Description Lattice Vector Quantizer (MDLVQ) has been an active area over an extended period of time and thereby addressed by many authors, *e.g.* [Vaishampayan et al., 2001; Diggavi et al., 2002; Goyal et al., 2002; Zhao, 2004; Diggavi et al., 2002; Østergaard et al., 2005]. The general unbalanced and asymmetric MDLVQ design was proposed by Diggavi, Sloane and Vaishampayan in [Diggavi et al., 2002]. Our scheme of MDC with conditional compression (MDC–CC), described in Section 4 employs this MDLVQ, described below.

The structure of the MDLVQ is shown in Figure 16.7. It operates as follows: a source vector is quantized to a lattice point $\lambda \in \mathbf{\Lambda}$. To send the information about λ over the two channels, a label function α is applied. We will from this point denote sublattice points with a subscript, *e.g.* $\lambda_{\mathbf{1}} \in \mathbf{\Lambda_1}$. The label function maps λ to a pair of sublattice points $(\lambda_{\mathbf{1}}, \lambda_{\mathbf{2}})$. We assume that the label function is one–to–one, so when both channels work, the inverse mapping $\alpha^{(-1)}$ will reconstruct the lattice point λ. Conversely, when only channel i works the reconstruction point of the source is the sublattice point $\lambda_{\mathbf{i}}$.

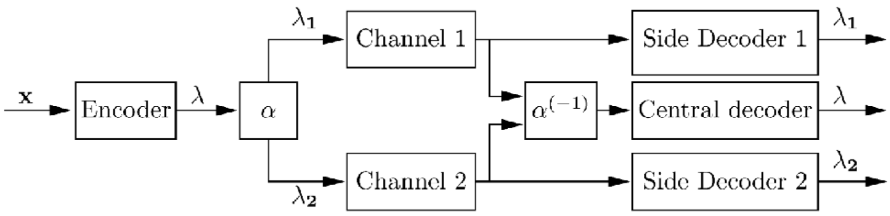

Figure 16.7. Block diagram of a two channel lattice vector quantizer.

To reduce the complexity of the label function α, the following three constraints on the MDLVQ are imposed:

- Constraint 1: The two sublattices $\mathbf{\Lambda_i}$ are geometrically similar and clean to $\mathbf{\Lambda}$, hence the reused index number N_i is given for $i \in 1, 2$, respectively.
- Constraint 2: There is a product sublattice $\mathbf{\Lambda_s}$ of $\mathbf{\Lambda_1} \cap \mathbf{\Lambda_2}$ that is geometrically similar to $\mathbf{\Lambda}$ and has reused index, $N_s = N_1 N_2$.
- Constraint 3: The label function must satisfy the shift-property, which means that $\alpha(\lambda + \lambda_{\mathbf{s}}) = \alpha(\lambda) + \lambda_{\mathbf{s}}, \forall \lambda \in \mathbf{\Lambda}, \lambda_{\mathbf{s}} \in \mathbf{\Lambda_s}$.

Encoding and decoding procedures for MDLVQ. The encoding procedure for an MDLVQ is a two step procedure, as illustrated in Figure 16.7. First the input vector $\mathbf{x}$ is quantized to the closed lattice point in $\mathbf{\Lambda}$,

$$\lambda = Q(\mathbf{x}). \tag{16.26}$$

Second, the label function maps the lattice point to the two sublattice points $\lambda_{\mathbf{i}}$, that are transmitted over the two channels:

$$(\lambda_{\mathbf{1}}, \lambda_{\mathbf{2}}) = \alpha(\lambda). \tag{16.27}$$

The two side decoding procedures are to find the best reconstruction point for a given λ_i. It is not guaranteed that the optimal reconstruction point is the sublattice point λ_i, as shown on Figure 16.7. This suboptimality induce a relatively small additional distortion, which may be neglected. Alternatively, one possible solution is to store a codebook containing reconstruction points, obtained by a Lloyd algorithm as investigated by [Zhao, 2004]. When both channels are working, the central decoding procedure is used as shown on Figure 16.7. The lattice point λ is found by the inverse label function $\alpha^{(-1)}$, hence $\lambda = \alpha^{(-1)}(\lambda_{\mathbf{1}}, \lambda_{\mathbf{2}})$.

Average distortion measurement. When both channels are working the inverse label function will reconstruct the lattice Λ, and we can therefore obtain the average central distortion from Eq. (16.10) as:

$$D_0 \approx G(\mathbf{\Lambda})\nu^{2/L}, \tag{16.28}$$

where ν is the volume of a Voronoi region. The average side distortion can be found by the distortion between input $\mathbf{x}$ and the sublattice point λ_i. By assuming that λ is the centroid of its Voronoi region and high resolution, it is shown in [Vaishampayan et al., 2001] that the side distortion can be expressed as:

$$D_i = D_0 + \sum_{\lambda \in \Lambda} \|\lambda - \alpha_i(\lambda)\|^2 P_\lambda, \tag{16.29}$$

where P_λ is the probability of the lattice point λ.

Rate for a MDLVQ. The entropy in bit per dimension for a MDLVQ assuming high resolution is given in [Diggavi et al., 2002],

$$R_0 \approx h(p) - \frac{1}{L}\log_2(\nu). \tag{16.30}$$

The entropy for a sublattice is derived in [Vaishampayan et al., 2001] and the entropy on each channel is found in [Diggavi et al., 2002] to be,

$$R_i \quad = \quad R_0 - \frac{1}{L}\log_2(N_i). \tag{16.31}$$

Before explaining how the label function is constructed, we will give a design example of an MDLVQ and illustrate the label function.

EXAMPLE 16.2 (AN EXAMPLE OF MDLVQ) *Let us make the simples and possible example of an asymmetric MDLVQ by using the A_2 or Z_2 lattices. To obey the similar and clean constraints for the MDLVQ design, we must determine two small numbers of N that are different to make the MDLVQ asymmetric ($N = 1$ is trivial). The smallest combination is obtained by Z_2 with $N_1 = 5$ and $N_2 = 9$. And for simplicity we choose the simplest generator matrix for Z_2,*

$$\mathbf{G} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{16.32}$$

As described in Section 2.0 the two generator matrices for the sublattices can be found:

$$\mathbf{G_1} = \begin{bmatrix} 2 & -1 \\ 1 & 2 \end{bmatrix}, \mathbf{G_2} = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}. \tag{16.33}$$

The product sublattice $\mathbf{\Lambda_s}$ can for this Z_2 example be found by:

$$\mathbf{G_S} = \mathbf{U_1 U_2 G} = \begin{bmatrix} 6 & -3 \\ 3 & 6 \end{bmatrix}. \tag{16.34}$$

The lattice and the three sublattices are shown on Figure 16.8, where we can verify that $N_1 = 5$, $N_2 = 9$ and $N_s = 45$. Furthermore, it can be realized when $\mathbf{\Lambda_s}$ is clean and we apply the shift–property that a label function describing the 45 lattice points in $V_s(0)$ can cover $\mathbf{\Lambda}$. The label function describing the 45

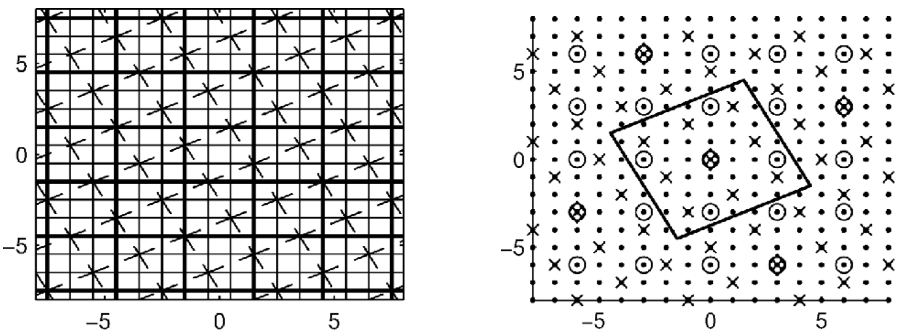

Figure 16.8. A MDLVQ example with $N_1 = 5$ and $N_2 = 9$, where the Voronoi regions are shown on the left and the centroids on the right. On the left, the solid line is the lattice $\mathbf{\Lambda}$, dashed line is the sublattice $\mathbf{\Lambda_1}$, wide line is the sublattice $\mathbf{\Lambda_2}$. On the right, the solid line is the Voronoi region $V_s(0)$.

lattice points are shown in Table 16.8. How to generate this label function will be explained in Section 2.0.0.

From the label function in Table 16.8 we can calculate the average distortions in the high resolution case from equations (16.28) and (16.29),

$$D_0 = \frac{1}{12}, \quad D_1 = 0.4833, \quad D_2 = 2.3500. \tag{16.35}$$

The rates for the example are

$$R0 = 2.0471, \quad R1 = 0.8861, \quad R2 = 0.4621. \tag{16.36}$$

Construction of the label function. Constructing the label function, a Lagrangian cost function is formulated which can be interpreted as the average

	$(-3,1)$	$(-2,-1)$	$(-1,2)$	$(-1,-3)$	$(0,0)$	$(1,3)$	$(1,-2)$	$(2,1)$	$(3,-1)$
$(-6,3)$	$(-3,1)$								
$(-6,0)$	$(-4,1)$	$(-3,-1)$							
$(-3,3)$	$(-3,2)$		$(-2,2)$						
$(-3,0)$	$(-3,0)$	$(-2,-1)$	$(-1,1)$		$(-1,0)$				
$(-3,-3)$		$(-2,-2)$		$(-1,-3)$					
$(-3,-6)$				$(-2,-3)$					
$(0,6)$			$(-1,3)$			$(0,3)$			
$(0,3)$			$(-1,2)$		$(0,1)$	$(1,3)$		$(2,2)$	
$(0,0)$	$(-2,1)$	$(-2,0)$	$(0,2)$	$(-1,-2)$	$(0,0)$	$(1,2)$	$(1,-1)$	$(1,1)$	$(2,-1)$
$(0,-3)$		$(-1,-1)$		$(0,-3)$	$(0,-1)$		$(0,-2)$		
$(0,-6)$				$(-1,-4)$			$(1,-3)$		
$(3,6)$						$(1,4)$			
$(3,3)$						$(2,3)$		$(2,1)$	
$(3,0)$					$(1,0)$		$(1,-2)$	$(2,0)$	$(3,0)$
$(3,-3)$							$(2,-2)$		$(3,-2)$
$(6,0)$								$(3,1)$	$(4,-1)$
$(6,-3)$									$(3,-1)$

Table 16.2. Label function α: The lattice points λ for a given λ_1 (rows) and λ_2 (column).

distortion, where γ_0 is the probability for using the central decoder, γ_i is the probability for using the i'th side decoder and γ_3 is the probability that both channels are not working. Using (16.29) the Lagrangian cost is,

$$J = \gamma_0 D_0 + \sum_{i=1}^{2} \gamma_i D_i + \gamma_3 \tag{16.37}$$

$$= (\gamma_0 + \gamma_1 + \gamma_2) D_0 + \sum_{i=1}^{2} \gamma_i \sum_{\lambda \in \Lambda} \|\lambda - \alpha_i(\lambda)\| P_\lambda + \gamma_3. \tag{16.38}$$

Constructing a good label function is to minimize the Lagrangian cost function. The central distortion and γ_3 can be neglected, since they are independent of the label function. In [Diggavi et al., 2002], the complexity is reduced by using constraint number 3, such that only the lattice points in $V_0 = \{\lambda \in \mathbf{\Lambda} : \lambda \in V_s(0)\}$ need to be assigned. Furthermore, assume the P_λ is constant over the V_0, such that the design objective is to minimize:

$$\sum_{\lambda \in V_0} (\gamma_1 \|\lambda - \alpha_1(\lambda)\| + \gamma_2 \|\lambda - \alpha_2(\lambda)\|). \tag{16.39}$$

The procedure for generating the label function as proposed in [Diggavi et al., 2002] is as follows:

1 Determine the indexes N_1,N_2, the lattice Λ, the sublattices Λ_1,Λ_2 and the Lagrangian multipliers γ_1 and γ_2. We will later give specific

examples and guidelines for how to determine these factors in the cooperative context.

2 Determine the two sets, $\eta_1 = V_0 \cap \mathbf{\Lambda_1}$ and $\eta_2 = V_0 \cap \mathbf{\Lambda_2}$, and the sets,

$$\zeta_i(\lambda_\mathbf{i}) \quad = \quad \{\lambda_\mathbf{j} \in \mathbf{\Lambda_j} : \lambda_\mathbf{j} \in V_0 + \lambda_\mathbf{i}\}, \quad \forall \lambda_\mathbf{i} \in \eta_i. \tag{16.40}$$

Determine all possible combinations of sublattice points by:

$$\epsilon_0 = \{(\lambda_\mathbf{i}, \lambda_\mathbf{j}) : \lambda_\mathbf{i} \in \eta_i, \lambda_\mathbf{j} \in \zeta_i(\lambda_\mathbf{j})\}, \tag{16.41}$$

where $(i, j) = (1, 2)$ or $(2, 1)$.

3 Matching the combinations in ϵ_0 to the lattice points in V_0, such that Eq. (16.39) is minimized, can be considered as a Mixed Integer linear Programming (MIP) problem. The optimization will result in two equivalent label functions for ϵ_0 when $i = 1$ and $i = 2$.

In [Diggavi et al., 2002], the authors first construct a label function that covers V_0 and subsequently extends the label function to the entire lattice using the shift–property. Hence, they can avoid quantization of the sublattice $\mathbf{\Lambda_s}$ in the encoding and decoding procedures. This will certainly require storage of a large label function or a mathematical description of the label function. In this chapter, for notational convenience, we will use the reduced label function, but the usage of the extended label function is straightforward.

EXAMPLE 16.3 (AN EXAMPLE OF MDLVQ (CONTINUED)) *Let us, finally, return to the MDLVQ example from Section 16.2 to show how the label function can now be generated by the procedure described in Section 2.0.0.*

In Step 1 all variables are known except $\gamma_1 = 0.0533$ and $\gamma_2 = 0.0438$. We will later explain how γ_i should be for a given loss probability. In step 2 we determine all the sublattice points in V_0 and

$$\begin{aligned} \eta_1 \quad &= \quad \{(-1,-3),(-2,-1),(-3,1),(1,-2),(0,0), \\ & \qquad (-1,2),(3,-1),(2,1),(1,3)\} \qquad (16.42) \\ \eta_2 \quad &= \quad \{(-3,0),(0,-3),(0,0),(0,3),(3,0)\}, \qquad (16.43) \end{aligned}$$

where η_1 and η_2 are shown as x-marks and circles on Figure 16.8(right), respectively. In step 3, we determine the sets ζ_1 and ζ_2. In step 4, with the optimization over all the elements in ϵ_0, we find the label function as shown in Table 16.8.

3. Optimizing Multiple Description Coding for losses in the Cooperative Context

The described MDLVQ can be optimized for the cooperative scheme shown on Figure 16.2. With an egoistic behavior, each participant in the cooperative scheme will demand that MDLVQ is optimized such that it gets the most out of the cooperation. This egoistic behavior maps into single terminal optimization of the MDLVQ. Subsequently, we will show that this optimization is not optimal for the MDLVQ cooperation scheme. A compromise among the participants in the cooperation can be to minimize the overall average distortion. We explain this in Section 3.0 .

The Single Terminal Optimization of the MDLVQ

Optimizing the MDLVQ in the cooperative scheme as shown on Figure 16.2 subject to either terminal 1 or terminal 2 will yield two different MDLVQ designs. In this section we first optimize to one of the terminals and then illustrate that this is not optimal for both of the terminals. Optimizing the MDLVQ design for terminal 1 is equivalent when optimizing to MDLVQ scheme when only one terminal is considered. The single terminal problem has been thoroughly analyzed in [Østergaard et al., 2004] for the high resolution case. The main results are outlined in the following. First, the side distortion for the MDLVQ is found in [Diggavi et al., 2002] to be,

$$D_i \approx \frac{\gamma_j^2}{(\gamma_i+\gamma_j)^2} G(\mathbf{\Lambda_s}) 2^{2h(p)} 2^{-2(R_1+R_2-R_0)}, \tag{16.44}$$

where $(i, j) = (1, 2)$ or $(2, 1)$ and γ_i is probability for receiving description i at terminal 1. The central distortion is given in Eq. (16.28),

$$D_0 \approx G(\mathbf{\Lambda})\nu^{2/L}. \tag{16.45}$$

Then, assuming an entropy constrain on the three channels, we note that $N_i\nu$ become a constant when combining Eq. (16.30) and Eq. (16.31),

$$N_i\nu = 2^{L(h(p)-R_i)} = c_i. \tag{16.46}$$

Combining Eq. (16.46) and Eq. (16.31) we get:

$$\frac{c_i}{\nu} = 2^{L(R_0-R_i)}. \tag{16.47}$$

Now, using Eq. (16.46) and Eq. (16.47), we can write the average distortion as:

$$\overline{D} = \gamma_0 G(\mathbf{\Lambda})\nu^{2/L} + \frac{\gamma_1\gamma_2^2+\gamma_2\gamma_1^2}{(\gamma_1+\gamma_2)^2} G(\mathbf{\Lambda_s})\left(c_1c_2\right)^{2/L}\nu^{-2/L} + \gamma_3. \tag{16.48}$$

Finally, we can determine the optimal volume ν by putting its derivative with respect to ν equal to zero. The solution leads to the optimal volume:

$$\nu = \left(\frac{\gamma_1\gamma_2}{\gamma_0(\gamma_1+\gamma_2)}\right)^{L/4}\left(\frac{G(\mathbf{\Lambda_s})}{G(\mathbf{\Lambda})}\right)^{L/4}\sqrt{c_1c_2}. \tag{16.49}$$

From Eq. (16.46) and Eq. (16.49) we determine the optimal N_i as follows.

$$N_i = \left(\frac{\gamma_0(\gamma_1+\gamma_2)}{\gamma_1\gamma_2}\right)^{L/4}\left(\frac{G(\mathbf{\Lambda_s})}{G(\mathbf{\Lambda})}\right)^{-L/4}\sqrt{\frac{c_i}{c_j}}. \tag{16.50}$$

This specifies an MDLVQ design optimized for one of the terminals. As we'll see in the following example, this design is however not always optimum for all involved terminals.

EXAMPLE 16.4 *Let us design an MDLVQ for a system with a unit variance Gaussian source, a Z_2 lattice and with a rate constraints $R_1 = 6$, $R_2 = 5.5$ and no rate constrain on the cooperative link. The loss probability on the three channels are $p_1 = 0.01$, $p_2 = 0.08$ and $p_c = 0.01$, and the three γ's for terminal 1 can be determined,*

$$\gamma_0 = 0.9017, \quad \gamma_1 = 0.0883 \quad \textit{and} \quad \gamma_2 = 0.0091. \tag{16.51}$$

Now, we determine the two constants c_i by Eq. (16.46), where the differential entropy of a unit variance Gaussian source is, $h(X) = \frac{1}{2}\log_2(2\pi e)$, such that the two constants are:

$$c_1 = 0.0042, \quad c_2 = 0.0083. \tag{16.52}$$

It becomes straightforward to find the optimal ν and N_i by Eq. (16.49) and Eq. (16.50):

$$\nu = 5.51 \cdot 10^{-4}, \quad N_1 = 7.6, \quad N_2 = 15. \tag{16.53}$$

In a similar manner, we can determine the loss probability for terminal 2:

$$\gamma_0 = 0.9017, \quad \gamma_1 = 0.0183 \quad \textit{and} \quad \gamma_2 = 0.0784, \tag{16.54}$$

and then determine the optimal ν and N_i with respect to terminal 2,

$$\nu = 7.39 \cdot 10^{-4}, \quad N_1 = 5.6, \quad N_2 = 11.3. \tag{16.55}$$

Note how the different loss probabilities, seen by the two terminals, lead to different quantizer designs as optimum for each of the two terminals.

From the example, we can conclude that sometimes optimization for each of the terminals is not feasible as they have to share the same encoder. In some cases with similar loss probabilities and because of cleanness and similarity constraints on the two sublatices the two MDLQ designs may indeed turn into the same MDLQ design, but in general egoistic behavior is suboptimal for some involved terminals.

Minimization of the Mean Distortion

A fair method to share the network resources can be to minimize the mean distortion over all terminals that cooperate. The mean distortion over all terminals is:

$$\begin{aligned}\overline{D} \;=\; & \frac{1}{2}\Big((\gamma_0^{(1)}+\gamma_0^{(2)})D_0+(\gamma_1^{(1)}+\gamma_1^{(2)})D_1+(\gamma_2^{(1)}+ \\ & \gamma_2^{(2)})D_2+(\gamma_3^{(1)}+\gamma_3^{(2)})\Big), \end{aligned} \tag{16.56}$$

where $\gamma_i^{(k)}$ is the γ_i for the $k-$th terminal. Therefore, we can determine the mean optimum MDLVQ design by adopting the interpretation that the system has only one terminal, with parameters:

$$\gamma_k = \frac{\gamma_k^{(1)}+\gamma_k^{(2)}}{2} \tag{16.57}$$

for $k \in \{0,1,2,3\}$, and then use the procedure described in Section 3.0. This approach generalize in a straight forward manner to any number of terminals. We illustrate this in the following continuation of the example.

EXAMPLE 16.5 ***[Continued]*** *Minimization of the mean distortion for the above example yields three new parameters:*

$$\gamma_0 = 0.9017, \quad \gamma_1 = 0.0533 \quad \textit{and} \quad \gamma_2 = 0.0438. \tag{16.58}$$

The optimal volume and reused index for the MDLVQ is,

$$\nu = 9.40 \cdot 10^{-4}, \quad N_1 = 4.4, \quad N_2 = 8.9. \tag{16.59}$$

From this, we can conclude that a good MDLVQ design can be $N_1 = 5$ and $N_2 = 9$. Note that this is the acutal design carried out in Section 2.0 when neglecting a scaling factor.

The rounding–off for the reused index N applied in the example is not described in the theory. However, the optimal N can be determined from the design of the possible N by evaluation of performance of each. Another method is to use an unstructured Multiple Description Vector Quantizer as described in [Koulgi et al., 2003] rather than the structured lattice, thus avoiding the lattice conditions on N.

Networks with Time–Varying Loss Probabilities

In practical cooperative networks, the exact loss probabilities for the channels in the cooperative network are not known at design time. Rather, these probabilities are known only with stochastic uncertainty, or they are known to be time–varying quantities during application of the coding system. In both cases, we can see the loss probability p_i and p_c as stochastic variables described by the pdf $f(p_i)$ and $f(p_c)$. The mean distortion over all terminals in a network with time–varying loss probabilities is given as follows.

$$\begin{aligned} \overline{d} &= \frac{1}{2}\iiint \Big((\gamma_0^1+\gamma_0^2)d_0+(\gamma_1^1+\gamma_1^2)d_1+(\gamma_2^1+\gamma_2^2)d_2+ \\ &\quad (\gamma_3^1+\gamma_3^2)\Big)\, f(p_1)f(p_2)f(p_c)dp_1dp_2dp_c. \end{aligned} \tag{16.60}$$

To minimize the mean distortion in the stochastic formulation we again determine the γ's,

$$\gamma_k = \frac{1}{2}\iiint \Big(\Gamma_c^1+\Gamma_c^2\Big) f(p_1)f(p_2)f(p_c)dp_1dp_2dp_c, \tag{16.61}$$

for $k \in \{0,1,2,4\}$, and then use the initial procedure described in Section 3.0. As an example of a cooperative network where stochastic and time–varying design is called for, we mention real–time media transmission using the real–time protocol (RTP). In this setting, the packet loss probabilities can be estimated at the receiver and fed back to the transmitter via the real–time control protocol (RTCP) in [Schulzrinne et al., 1996]. For this type of applications, we can design a bank of MDLVQ's, such that the encoder can select the most suitable MDLVQ design for the given loss probability. This bank construction is investigated in [Larsen et al., 2005] where a significant gain was found with increasing the number of designs in the bank.

4. MDC with Conditional Compression (MDC–CC)

As explained in the previous sections, a MD scheme introduces coding overhead in order to provide self–sufficient descriptions. Consider the case with two descriptions, d_1 and d_2. Each description is carrying information that is sufficient to obtain a low–quality replica of the original source information. This means that there is some portion in each of d_1 and d_2 that is carrying identical information about the source. Assume that d_1 has been received at the destination. Then it is not necessary that the full d_2 is sent to the destination, but only the information from d_2 that is not contained in d_1. We denote this information as $(d_2|d_1)$ and call it a conditional description provided that d_1 is available.

An important consequence of this concept is that the compression of the source information is not performed at the source, but in a node that lies on the path between the source and destination. This concept is shown on Figure 16.9. The source node S produces two descriptions and sends d_1 and d_2 through two disjoint paths, passing through X and Y, respectively. Assume that the feedback from destination D is not timely available at S, but it is available at X or Y. Now let d_1 arrive through X at the destination D, then D can inform Y about this through the fast feedback channel. With such an information, Y can re–code d_2 and send only the conditional information $(d_2|d_1)$. Hence, the information that traverses the path from Y to D is compressed, as one example of this solution D co–insides with X or Y and X and Y are cooperating terminals. In the sequel we describe the proposed realization of the MDC–CC in the MDLVQ framework.

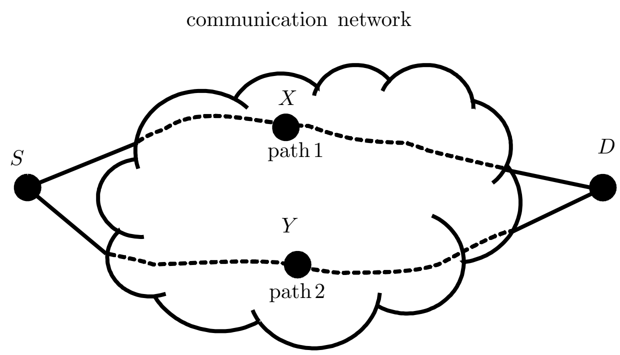

Figure 16.9. A communication network that provides two paths from S to D. X and Y are intermediate nodes along the paths. The feedback from D is not available timely at S, but can be available at X or Y.

MDLVQ for MDC–CC

We start by making an interpretation of the MDLVQ, inspired by the design algorithm described in Section 2. In the design algorithm of the MDLVQ, the lattice points in V_0 are assigned by the label function and, subsequently, the label function is expanded to R^L. In this new scheme, we keep the reduced label function for V_0 and reintroduce the pre–encoder $\lambda_s = Q_s(X)$, equivalent to the shift–property. We transmit λ_s (neglecting the small offset in λ_s^+) over both channels and the relative refinement information λ_1^* and λ_2^* over each channel, instead of transmitting the λ_1 and λ_2 over each channel, as shown on Figure 16.10. Regarding the decoding, when only description i is received, then the reconstruction point is based on information from $\lambda_\mathbf{s}$ and $\lambda_\mathbf{i}^*$. Conversely, when both descriptions are received the reconstruction is based on $\lambda_\mathbf{1}^*$, $\lambda_\mathbf{2}^*$ and $\lambda_\mathbf{s}$ or $\lambda_\mathbf{s}^+$. Clearly, when applying the central decoder, λ_1^* is a conditional description provided that λ_s^+ and λ_2^* are received, as denoted $(d_1|d_2)$ in last section. The interpretation for λ_2^* is analogous. In the next two subsections we will describe how to construct the new relative label function β, and furthermore it will be shown that the entropies in the MDLVQ interpretation are maintained.

Clearly the distortions are maintained since the interpretation has no impact on the λ and λ_1 and λ_2, at the decoders.

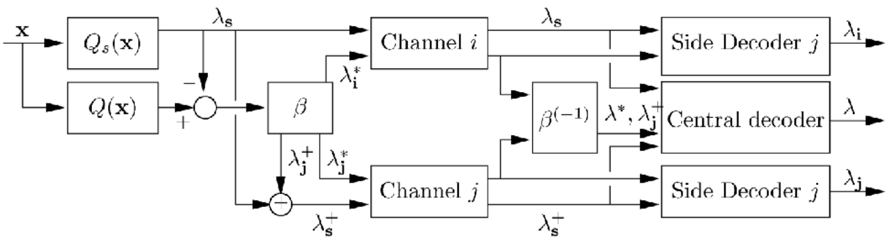

Figure 16.10. An interpretation of a MDLVQ.

Construction of the relative label function. In Section 2.0.0 where the label function was constructed, all combinations of the sublattice points η_i and the sublattice points in ζ_i were considered. By definition, all sublattice points in η_i are included in V_0, but the sublattice points in ζ_i belong to a larger set. Each sublattice point $\lambda_j \in \zeta_i$ can be described by a relative sublattice point, $\lambda_j^* \in \eta_j$ and a corresponding offset sublattice point λ_j^+. The relationship between the $\lambda_j \in \zeta_i$ and the λ_j^* is:

$$\lambda_j^* = \lambda_j + \lambda_j^+, \tag{16.62}$$

where $\lambda_j^+ \in \Lambda_s$. It is built–in the optimization in Step 3, that a combination of a sublattice point in η_i and a sublattice point in η_j is only used once, to obey the shift–property. Thus, we can after designing the label function compress the label function to only include combinations of sublattice points from η_i and η_j, without any conflict. For simplicity, we also denote the sublattice points $\lambda_i^* \in \eta_i$. Thus, we can construct a relative label function, β, that maps a lattice point $\lambda \in V_0$ to λ_i^*, λ_j^* and λ_j^+, as shown on Figure 16.10.

Example 16.6 (An Example of MDLVQ (Continued)) *Let us return to the MDLVQ example from Example 16.2 to show how the label function α can be reformulated to a relative label function β, where $i = 1$ and $j = 2$. For each lattice point λ in Table 16.8 the corresponding sublattice point λ_j is reduced to λ_j^*, will result in Table 16.3. Furthermore, for each lattice point λ, the difference between λ_j and λ_j^* will result in λ_j^+, as shown on Table 16.4. The label function β is completely described by Table 16.3 and 16.4. From Table 16.3 we note that all the combinations of the sublattice points in η_1 and η_2 are used, which will be used in next section.*

	$(-3,0)$	$(0,3)$	$(0,0)$	$(0,-3)$	$(3,0)$
$(-3,1)$	$(-3,0)$	$(-4,1)$	$(-2,1)$	$(-3,2)$	$(-3,1)$
$(-2,-1)$	$(-2,-1)$	$(-3,-1)$	$(-2,0)$	$(-1,-1)$	$(-2,-2)$
$(-1,2)$	$(-1,1)$	$(-1,2)$	$(0,2)$	$(-2,2)$	$(-1,3)$
$(-1,-3)$	$(-1,-4)$	$(-2,-3)$	$(-1,-2)$	$(0,-3)$	$(-1,-3)$
$(0,0)$	$(-1,0)$	$(0,1)$	$(0,0)$	$(0,-1)$	$(1,0)$
$(1,3)$	$(2,3)$	$(1,3)$	$(1,2)$	$(1,4)$	$(0,3)$
$(1,-2)$	$(1,-3)$	$(2,-2)$	$(1,-1)$	$(0,-2)$	$(1,-2)$
$(2,1)$	$(2,1)$	$(2,2)$	$(1,1)$	$(3,1)$	$(2,0)$
$(3,-1)$	$(3,-1)$	$(3,-2)$	$(2,-1)$	$(4,-1)$	$(3,0)$

Table 16.3. Label function β: The relative lattice points λ^* for a given λ_1^* (column) and λ_2^* (rows).

	$(-3,0)$	$(0,3)$	$(0,0)$	$(0,-3)$	$(3,0)$
$(-3,1)$		$(-6,-3)$		$(-3,6)$	$(-9,3)$
$(-2,-1)$		$(-6,-3)$			$(-6,-3)$
$(-1,2)$				$(-3,6)$	$(-3,6)$
$(-1,-3)$	$(3,-6)$	$(-3,-9)$			$(-6,-3)$
$(0,0)$					
$(1,3)$	$(6,3)$			$(3,9)$	$(-3,6)$
$(1,-2)$	$(3,-6)$	$(3,-6)$			
$(2,1)$	$(6,3)$			$(6,3)$	
$(3,-1)$	$(9,-3)$	$(3,-6)$		$(6,3)$	

Table 16.4. Label function β: The offset lattice points λ_2^+ for a given λ_1^* (column) and λ_2^* (rows). On the empty places the offset lattice point is zero.

Rate computation for a MDC–CC in the MDLVQ framework. To determine the entropies in the MDC-CC framework we will assume high resolution, as in Section 2.0. The entropy of a sublattice is derived in [Vaishampayan et al., 2001] for a given reused index and assuming high resolution. Thus, we can realize that the entropy for the productive sublattice is,

$$R_s = R_0 - \frac{1}{L}\log_2(N_1 N_2). \tag{16.63}$$

To determine the entropy of the offset lattice point λ_s^+, we first realize the following: That λ_s and λ_s^+ is adjacent (in space) and therefore the $Pr(\lambda_s) \approx Pr(\lambda_s^+)$, when assuming high resolution. Thus the entropy of λ_s^+ is equal to the entropy of λ_s. A similar approximation for λ_i is taken in [Vaishampayan et al., 2001], when calculating the side distortion. Another way to see this is, when λ^* is close to the centroid of V_0 then λ_j^+ is 0. Conversely, when λ^* is close to the boundary of the Voronoi region of V_0 then λ_j^+ is not-zero. Which was also the case in Table 16.4. So, when assuming high resolution, then the cardinality of η_j is large and the probability for λ^* is close to the boundary goes towards 0. Thus, the probability for $\lambda_j^+ \neq 0$ goes towards 0 and thereby the entropy of λ_s^+ goes towards the entropy of λ_s.

Applying the label function β it is guaranteed that all the combination of $\lambda_i \in \eta_i$ and $\lambda_j \in \eta_j$ is used and only once. Furthermore, when assuming high resolution, this implies equal probability for all $\lambda_j \in V_0$, enables us to determine the entropy of the refinement information and thereby the entropy of the conditional compression,

$$R_j^* = \frac{1}{L}\log_2(N_i). \tag{16.64}$$

We can then verify that the entropy is maintained, by $R_s + R_j^*$ and compared with R_j from Section 2.0.0. On the other hand, for low resolution it can also be argued that the entropy on the channel is maintained. Because, the information in λ_j from the classical MDLVQ scheme is exactly the same information in λ_s^+ combined with λ_j^* in the MDC–CC scheme. Where conversion between them can always be performed by ether a quantization $\lambda_s^+ = Q_s(\lambda_j)$ or with a careful design of the entropy coder that keeps the information about λ_j^* and λ_s^+ separated. The interpretation for the entropy of the channel i is analogous.

Encoding and decoding procedure MDC–CC in the MDLVQ framework. The encoding procedure for MDC–CC with MDLVQ is a three step procedure, as illustrated in Figure 16.10. First the input vector $\mathbf{x}$ is quantized to the closest sublattice point in $\mathbf{\Lambda_s}$,

$$\lambda_{\mathbf{s}} = Q_s(\mathbf{x}). \tag{16.65}$$

The second step, is to quantize the input vector to the closest lattice point λ in $\mathbf{\Lambda}$,

$$\lambda = Q(\mathbf{x}). \tag{16.66}$$

In the third step, the sublattice point $\lambda_\mathbf{s}$ is subtracted from the lattice point λ. This ensures that $\lambda' = \lambda - \lambda_\mathbf{s}$ is included in V_0 and the label function β can be applied, similar to the shift–property in the design algorithm. Applying the label mapping, we get the two relative sublattice points and the offset sublattice point,

$$(\lambda^*_\mathbf{i}, \lambda^*_\mathbf{j}, \lambda^+_\mathbf{j}) = \beta(\lambda'). \tag{16.67}$$

Finally, the two multi–points are transmitted over the two channels:

$$\text{Channel } i : (\lambda_\mathbf{s}, \lambda^*_\mathbf{i}), \quad \text{Channel } j : (\lambda^+_\mathbf{s}, \lambda^*_\mathbf{j}). \tag{16.68}$$

The side decoding procedures are simply the sum of the multi-points, $\lambda_\mathbf{i} = \lambda_\mathbf{s} + \lambda^*_\mathbf{i}$ for side decoder i and $\lambda_\mathbf{j} = \lambda^+_\mathbf{s} + \lambda^*_\mathbf{j}$ for side decoder j. When both channels are working, the decoding procedure is a two–step procedure, as shown on Figure 16.10. The first step is to find the relative lattice point λ^* and the offset lattice point $\lambda^+_\mathbf{j}$ by applying the inverse label function, $(\lambda^*, \lambda^+_\mathbf{j}) = \beta^{(-1)}(\lambda^*_\mathbf{1}, \lambda^*_\mathbf{2})$. Subsequently the relative lattice point is added to the sublattice $\lambda_\mathbf{s}$ in order to reconstruct the lattice point, when λ_s is available. Otherwise, when λ^+_s is available the lattice point is reconstructed by: $\lambda = \lambda^+_\mathbf{s} - \lambda^+_\mathbf{j} + \lambda^*$.

EXAMPLE 16.7 *Let us now revise the the cooperative scheme from Figure 16.2, where two terminals exchange information over the cooperative link. Let terminal 1 receive d_1, which contains $\lambda_s + \lambda^*_1$. In order to obtain the high quality only the relative refinement information λ^*_2 needs to be transmitted over the cooperative link, since terminal 1 already knows λ_s. For this scheme with conditional compression to work, the terminal 2, before compressing the description d_2, must be sure that terminal 1 has received d_1. Hence, a full cooperative protocol operates with adaptation to the feedback: if terminal 2 receives positive feedback that terminal 1 received d_1, then compressed description is forwarded. Otherwise, with negative feedback or no feedback at all, terminal 2 forwards the full description d_2.*

It can be observed that, regarding the source coding, the network is an *active actor* when MDC–CC is used. Compared to this, in case of conventional MDC or LC, the network is a passive actor and only the source node does the source coding and compression. We can say that such operation of MDC–CC is a representative case of a cross–layer optimization in the protocol design.

5. Discussion

Having presented the MDC basics as well as specific MDC optimizations and schemes suitable for cooperative communications, in this section we will discuss three generic scenarios in which source coding based on MDC appears as a suitable solution within the paradigm of cooperative networking.

- Data delivery with Cooperative Sources (CS–scenario)
- Data delivery with Cooperative Destinations (CD–scenario)
- Data delivery with Meshed Cooperation (MC–scenario)

We will see that the overall efficiency of those cooperative scenarios naturally increases when MDC–CC is applied.

Data delivery with Cooperative Sources (CS–scenario)

In this scenario the whole network can be considered as a distributed source of information for the destination node D. Therefore, the data transmission can take advantage of the diversity provided by the network, such as path diversity. We illustrate this scenario through two examples.

Example CS–1. This scenario has been depicted on Figure 16.9. The nodes X and Y can be considered as distributed sources of a correlated information, since the descriptors forwarded by them are correlated. If X and Y are not cooperating, then each of them is "blindly" forwarding the appropriate description generated at S. If X and Y are mutually coordinated in transmitting the data to D, then they can be considered as cooperative sources of information. As we have explained in Section 4, this cooperation is realized through the use of MDC–CC. The cooperation between X and Y can be initiated by the destination through the feedback paths or via a link between X and Y. This scenario sets a stage for building protocols for cooperative streaming. As a special case, the nodes X and Y can be two wireless access points (APs) and the destination node D can be a terminal which lies in the radio range of both APs.

Example CS–2. The distributed storage has been outlined as an appropriate application for MDC in [Goyal, 2001]. For example, such is the case where a multimedia content is stored on several locations with multiple description (MD) encoding, such that each location (content server) contains a single descriptor pertained to the video stream. Consider the example on Figure 16.11. With a conventional MDC, the user would require a whole description from each content server. If MDF–CC is utilized, then the user needs to get the full description d_2 from the server S_2, while it retrieves the conditionally compressed descriptions $(d_1|d_2), (d_3|d_2)$ from the servers S_1, S_3. This decreases the overall traffic in the network. Again, in this case the cooperation among the servers can be initiated by feedback from the user or through the usage of direct coordination among S_1, S_2, S_3. Note that layered coding cannot provide such a forwarding mechanism in this scenario because the server that should provide

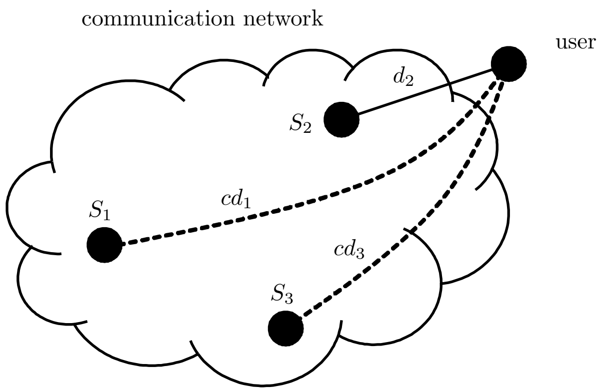

Figure 16.11. Access to multimedia content that is stored by MD encoding in three content servers S_1, S_2 and S_3. The user gets a full description from the closest server (d_2 from S_2), while it retrieves the compressed descriptions cd_1, cd_3 from the other two servers.

the full description is not predefined. That is, if another user is close to S_3, then the compressed descriptions should come from S_1 and S_2.

Data delivery with Cooperative Destinations (CD–scenario)

In these scenarios the source data is broadcasted to several destination terminals, while the terminals use the communication links among them to cooperate and thus enhance each other's reception of the broadcasted data.

Example CD–1. Figure 16.12 illustrates a broadcast scenario in which the feedback link from the terminals MS_1, MS_2 to the source base station BS is not available. The BS encodes the information with two descriptions d_1, d_2 and transmits them over the air. Depending on what has been received at each terminal, the cooperative link between the terminals is used for MS_1 to transmit a whole or compressed description to the terminal MS_2 and vice versa. In this case the MDC–CC scheme reduces the traffic on the cooperative link. Furthermore, MDC–CC appears to be an essential ingredient of this scenario. If the capacity of the cooperative link is less than the capacity of the broadcast transmission.

Example CD–2. To illustrate this scenario we can again use Figure 16.12, but in this case we assume that the links from each terminal to the BS are bi directional. The usage of multiple descriptions enables dynamic compression and routing of the broadcasted information. This can mean, for example, that only the description that is lost is forwarded through the cooperative link. The BS should initially transmit both full descriptions, but upon request for

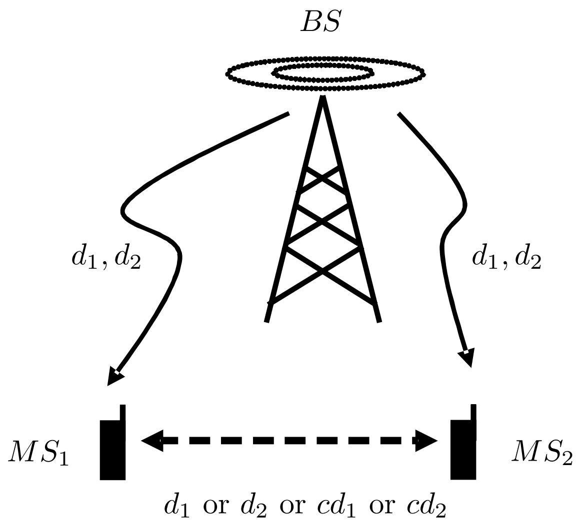

Figure 16.12. Broadcast scenario with cooperative destinations (MS_1 and MS_2) when the link from the source (base station BS) is unidirectional such that feedback to the source is not available.

retransmission from MS_1 and/or MS_2 the compressed descriptions can be provided by either BS or the other MS.

Data delivery with Meshed Cooperation (MC–scenario)

In this scenario each node involved in the communication can be a source of information and a destination. Such can be the case of video–conferencing or gaming. The source information of $node_1$ is encoded by two descriptions d_{12} and d_{13} and they are sent through the links l_{12} and l_{13}, respectively. Similarly, $node_2$ sends d_{21}, d_{23} through l_{21}, l_{23}, and $node_3$ sends d_{31}, d_{32} through l_{31}, l_{32}. Depending on the link conditions, each node can forward compressed or full description on behalf of another node. For example, if there are no errors, $node_1$ can forward the compressed descriptor cd_{31} to $node_2$, such that $node_2$ is able to completely reconstruct the source information of $node_3$, since it receives d_{32} through l_{32}. However, if the link conditions on l_{32} become bad, then $node_1$ forwards the full description d_{31}, while the quality at $node_2$ degrades gracefully.

6. Conclusion

In this chapter, we have presented ideas that relate the paradigm of cooperative communications to the problems of source encoding and compression.

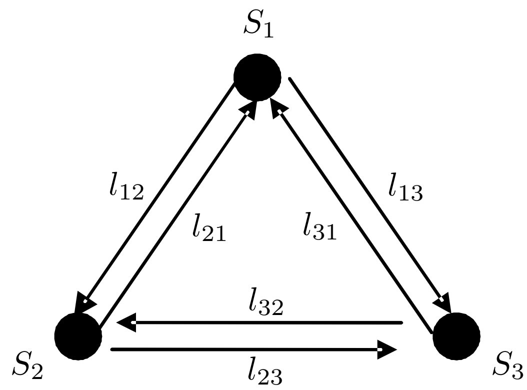

Figure 16.13. Meshed cooperation with three nodes, where each node is a source and a destination of information. l_{ij} denotes unidirectional link between nodes i and j.

The possibilities for conducting cooperative communication can be vastly expanded when the communication protocols use the features of the transmitted data in terms of source encoding. Our starting point is represented by the class of Multiple Description Coding (MDC) methods, in which multiple descriptions are produced information. For balanced MDC, each description is sufficient to restore the source information with certain rate distortion and the rate distortion is decreased as more descriptions of the same source symbol are received and used in the decoding. This chapter brings three distinctive contributions. The *first contribution* is related to the multiple description lattice vector quantizer (MDLVQ), which is a practical procedure for MDC where the quantizer is highly geometrically structured. We have shown how to optimize the design of MDLVQ in the cooperative context. The *second contribution* is a proposal of a novel MDC scheme, termed MDC with Conditional Compression (MDC–CC). This scheme emerges from the joint consideration of the source encoding and the networking and, although general, we elaborate MDC–CC in case of two descriptions, d_1 and d_2 per source information. The basic observation is that, once a node X in the network that contains description d_2 has the information that the destination already has received d_1, then it can compress the description d_2 before forwarding it to the destination. It can be observed that the concept of MDC–CC can move the compression task at any node in the network instead of solely the source node. We also introduce implementation of MDC–CC based on MDLVQ. Finally, the *third contribution* introduces several scenarios for cooperative communication in which the features of MDC–CC can boost the performance of the cooperative scheme. We also provide a taxonomy for the cooperative scenarios based on MDC. This taxonorny exposes a field of open questions on the MDC–CC concept. Answering a few of these questions are topics of our currect research.

References

Conway, J. and Sloane, N. (1982a). Fast quantizing and decoding and algorithms for lattice quantizers and codes. *Information Theory, IEEE Transactions on*, 28(2):227–232.

Conway, J. and Sloane, N. (1982b). Voronoi regions of lattices, second moments of polytopes, and quantization. *Information Theory, IEEE Transactions on*, 28:211–226.

Conway, J. H., Rains, E. M., and Sloane, N. J. A. (1999). On the existence of similar sublattice. *Canadian J. Math.*, 51:1300–1306.

Conway, J. H. and Sloane, N. J. A. (1999). *Sphere Packings, Lattices and Groups*. Springer-Verlag.

Cover, Thomas M. and Thomas, Joy A. (1991). *Elements of Information Theory*. John Wiley.

Diggavi, S. N., Sloane, N. J. A., and Vaishampayan, V. A (2002). Asymmetric multiple description lattice vector quantizers. *Information Theory, IEEE Transactions on*, 48(1):174–191.

Goyal, V. K (2001). Multiple description coding: compression meets the network. *Signal Processing Magazine, IEEE*, 18:74–93.

Goyal, V. K., Kelner, J. A., and Kovacevic, J. (2002). Multiple description vector quantization with a coarse lattice. *Information Theory, IEEE Transactions on*, 48:781–788.

Gray, Robert M. (1990). *Source Coding Theory*. Kluwer Academic Publishers.

Gupta, P. and Kumar, P. R. (2000). The capacity of wireless networks. *Information Theory, IEEE Transactions on*, pages 388–404.

Koulgi, P., Regunathan, S. L., and Rose, K. (2003). Multiple description quantization by deterministic annealing. *Information Theory, IEEE Transactions on*, 49:2067–2075.

Larsen, M. H., Arnbak, K. D., and Andersen, S. V. (2005). Optimization of multiple description quantizers for stochastic and time-varying loss probabilities. *IST 2005*.

Nosratnia, A., Hunter, T. E., and Hedayat, A. (2004). Cooperative communication in wireless networks. *IEEE Communications Magazine*, pages 74–80.

Østergaard, J., Jensen, J., and Heusdens, R. (2004). Entropy constrained multiple description lattice vector quantization. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 4.

Østergaard, J., Jensen, J., and Heusdens, R. (2005). n-channel symmetric multiple-description lattice vector quantization. In *IEEE Proc. Data Compr. Conf.*, pages 378–387.

Schulzrinne, H., Casner, S., Frederick, R., and Jacobson, V. (1996). RFC 1889: Rtp: A transport protocol for real-time applications.

Sergio, S. D., Vaishampayan, V. A., and Sloane, N. J. A. (1999). Multiple description lattice vector quantization. *Data Compression Conference (DCC)*.

Sloane, N. J. A. (2005). The on-line encyclopedia of integer sequences.

Vaishampayan, V. A. (1993). Design of multiple description scalar quantizers. *Information Theory, IEEE Transactions on*, 39(3):821–834.

Vaishampayan, V. A. and Domaszewicz, J. (1994). Design of entropy-constrained multiple-description scalar quantizers. *Information Theory, IEEE Transactions on*, 40:245–250.

Vaishampayan, V. A., Sloane, N. J. A., and Servetto, S.D. (2001). Multiple-description vector quantization with lattice codebooks: design and analysis. *Information Theory, IEEE Transactions on*, 47:1718–1734.

Zhao, D. Y. and Kleijn, W. B. (2004). Multiple-description vector quantization using translated lattices with local optimization. *Global Telecommunications Conference (GLOBECOM)*, 1:41–45.