國立交通大學

電信工程學系

碩士論文

多重描述量化碼書的索引指定設計

Index Assignment for Multiple Description Quantization over Mobile Ad Hoc Networks

研究生：林宜德

指導教授：張文輝 博士

中華民國九十五年六月

多重描述量化碼書的索引指定設計

Index Assignment for Multiple Description Quantization over Mobile Ad Hoc Networks

研 究 生：林宜德　　　　Student: I-Te Lin

指導教授：張文輝　　　　Advisor: Wen-Whei Chang

國立交通大學

電信工程學系

碩士論文

A Thesis

Submitted to Department of Communication Engineering

College of Electrical and Computer Engineering

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in Communication Engineering

June 2006

Hsinchu, Taiwan, Republic of China

中華民國 九十五年 六月

多重描述量化碼書的索引指定設計

研究生：林宜德　　　　　　　　指導教授：張文輝 博士

國立交通大學電信工程學系

摘要

本論文探討多重描述量化碼書之索引指定最佳化問題，進一步應用在無線隨意網路環境之分散式語音辨識系統。在系統的傳送端，多重描述量化器將語音參數量化，並根據事先定義的碼書索引指定產生兩個或多個描述，這些描述再通過多個互相獨立且無記憶性的通道傳送，在接收端則以一個符合最小期望擾動的解碼器將語音參數還原。傳統解決碼書索引指定最佳化的問題，是利用多重描述二進位置換演算法，它的做法是不斷地置換一對碼字的位置直到期望擾動收斂為止。為了要降低其運算複雜度，我們將索引指定設計轉換成符合線性規劃的雙方相配問題，並且提出多重描述亨格利演算法，藉此快速的建立最佳化的碼書索引指定。系統模擬是在不同的網路封包漏失環境中，利用國語數字串的分散式語音辨識進行碼書索引指定的性能評估。實驗結果顯示在隨機漏失和吉伯爾模型漏失的環境下，所提出的多重描述傳輸機制比其它單一描述傳輸方法具有更強健的性能表現，同時也利用無線隨意網路模擬平台評估分散式語音辨識的性能。

Index Assignment for Multiple Description Quantization over Mobile Ad Hoc Networks

Student: I-Te Lin Advisor: Dr. Wen-Whei Chang

Department of Communication Engineering
National Chiao Tung University

ABSTRACT

This study addresses the index assignment optimization for multiple description quantizers and its application to distributed speech recognition over Mobile Ad Hoc Networks (MANET). In the encoder, the speech parameters are quantized and mapped to two or more descriptions according to a predefined index assignment. After being transmitted over multiple independent channels, the decoder uses the received descriptions to reproduce the speech parameters. The optimization criterion and a practical approach based on the multiple description binary switching algorithm (MD-BSA) are first presented for the index assignment optimization. The basic idea of the MD-BSA is to switch a pair of codevectors recursively until the expected channel distortion can not be further reduced. In order to reduce its computational complexity, we formulated the index assignment problem on the basis of a linear programming framework and then proposed a fast local search algorithm based on the multiple description Hungarian algorithm (MD-HA). Experiments on the Mandarin digit string recognition task show that the proposed multiple description scheme outperforms single description methods in the presence of random and Gilbert-model packet losses. The ns-2 based MANET simulation was also conducted to examine the performances of the proposed multiple description transmission scheme.

Acknowledgements

I would like to acknowledge my advisor, Prof. Wen-Whei Chang, for his valuable guidance and constant support throughout my graduate studies at NCTU. I would also like to thank Tai-Kuei Fu for his help in conducting the Mandarin digital string recognition task. Moreover, I am grateful to my colleagues in the Speech Communication Laboratory for their assistance, discussions, and suggestions. Finally, I deeply appreciate my parents and friends for their encouragement and understanding.

Contents

Abstract (in Chinese) i

Abstract ii

Acknowledgements iii

Contents iv

List of Tables vi

List of Figures vii

1 Introduction 1

2 NS-2 Based MANET Simulation 4

2.1 Routing Protocols for the MANET 5

2.1.1 Destination-Sequenced Distance Vector (DSDV) 5

2.1.2 Ad-hoc On-Demand Distance Vector (AODV) 6

2.2 NS2 Simulation Platform 6

2.2.1 Wireless Multihop Simulation Environments 6

2.2.2 Simulation Methodology 7

2.2.3 Trace File Analysis 9

2.3 A Simulation Example 11

3 MDVQ Transmission Scheme 14
3.1 System Model 14
3.2 Optimal MDVQ Decoder 15

4 Index Assignment Algorithms 19
4.1 Quality Criterion 20
4.2 Multiple Description Binary Switching Algorithm 21
4.3 Bipartite Matching Problem and Hungarian Algorithm 23
4.4 Multiple Description Hungarian Algorithm 31

5 Experimental Results 34
5.1 ETSI DSR Framework 35
5.2 A Preliminary Experiment 36
5.3 Channel-mismatch Problem 36
5.4 DSR over Symmetric Channels with Random Packet Losses 38
5.5 DSR over Symmetric Gilbert Channels 39
5.6 DSR over Mobile Ad Hoc Networks 41

6 Conclusions and Future Work 45

Bibliography 47

List of Tables

2.1 Simulation parameters for evaluating the performances of the split VQ (SVQ) with a single description. 12

4.1 Effects on the entries in the reduced matrix. 30

5.1 Various test conditions. 38

5.2 DSR feature pairs and associated bit allocation. 39

5.3 Gilbert-model loss conditions. 41

5.4 Simulation parameters for evaluating the performances of the proposed MDVQ. 43

List of Figures

2.1 Packet delivery fractions of the DSR transmission pair competed with various numbers of CBR sources. 13

3.1 MDVQ transmission scheme. 15
3.2 The decoding process of the side decoder. 17

4.1 Two possible ways of switching codevectors. 22
4.2 Network structure of the bipartite matching problem. 24
4.3 Bipartite matching matrix. 26
4.4 Determining the reduced matrix. 27
4.5 An optimal bipartite matching. 28
4.6 Original and reduced matrix. 28
4.7 Covering all zeros with $k = 2$ lines. 29
4.8 Modified matrix and optimal bipartite matching. 30

5.1 SNR performances of MDVQ with various index assignment algorithms. 37
5.2 SNR performances of MD-BSA and MD-HA for different test conditions. 38
5.3 Recognition performances of various coding schemes in the presence of random packet losses. 40
5.4 Gilbert channel model. 41
5.5 Recognition performances of various coding schemes in the presence of Gilbert-model packet losses. 42

5.6 Recognition performances of the proposed MDVQ using various numbers of CBR sources. 44

Chapter 1

Introduction

Mobile ad hoc networks (MANET) is a new networking technology that provides information exchange in a group of wireless mobile hosts. The main feature of the MANET is to exploit the routing protocol based hop-by-hop transmission methodology to establish a communication link when individual host has its own moving speed and direction. This scenario is practical in new applications such as the intelligent transportation system (ITS) or the advanced traveller information system (ATIS). For example, a driver on the highway may use the ATIS to acquire his favorite restaurant information at the next rest area. The distributed speech recognition (DSR) front-end in the car extracts speech parameters which are then transmitted over the MANET to the back-end recognition server. After recognizing the speech signal, the server sends the restaurant information, which may be a short vedio introducing the menu, back to the driver. The driver may again use this system to order his meal in advance. Obviously, the whole information access procedure is dominated by real time multimedia communications. Thus, in order to provide high quality multimedia service to consumers, robust transmission schemes and information processing must be adopted to ensure that the overall system performance meets the quality of service (QoS) requirements.

For transmitting multimedia over packet-based networks such as the MANET, it is essential to deal with packet loss, end-to-end delay, and delay variation (jitter) to meet QoS requirements. Packet loss problem is usually alleviated by packet-level foward error correction (FEC) based on parity codes and Reed Solomon codes [1][2]. Packet-level FEC generates and sends some redundant information by exploiting previously transmitted multiple packets. However, for interactive audio applications such as VoIP and audio conferencing, which require high constraints on end-to-end delay, this technique is not applicable because of the additional delay caused by waiting for the redundant information at the decoder. Besides, bursty packet loss significantly degrades the performance of the FEC.

Recently, there has been much attention focused on using a novel coding scheme, multiple description (MD) quantization, to combat the packet loss problem. The basic idea of this scheme is to exploit the largely uncorrelated loss and delay behavior of multiple independent transmission paths. For example, a two-description encoder encodes the information source into two descriptions and transmits them over two independent channels. In the decoder, even if only one description is received, the information source can still be reconstructed but with a reduced quality. A higher quality reconstruction can be achieved if both descriptions are received.The research aspects of multiple description quantization can be categorized into two parts: 1) the optimal encoder and decoder design, 2) the optimization of the index assignment. The work in [3] discussed the optimization criterion for the encoder and decoder, the rate distortion bound, and the index assignment methodology for multiple description scalar quantizers (MDSQ). Particularly, a generalization of Lloyd's algorithm is developed to find out the optimal encoder and decoder. By minimizing the index spread parameter, two diagonal index assignments, called the nested index assignment and the linear index assignment, are proposed. Channel optimized quantizer design was addressed in [4] and [5]. A discrete memoryless channel model that considers both erasures and symbol errors is proposed to optimize the encoder and decoder in [4]. Also proposed is Gilbert's two-state channel model [5], where time-division multiplexing or packet interleaving is used to characterize channel erasure errors. In [6], a simulated annealing based method is adopted to optimize the index assignment. Other contributions from [7], [8], and [9] address the index assignment problem based on lattice vector quantizers.

In some practical applications such as VoIP or distributed speech recognition, quantizer codebooks have been standardized and therefore most works will focus on the index assignment and the decoder design. For a given quantizer codebook with a single description [10], a locally optimum index assignment algorithm, called the Binary Switching Algorithm (BSA), was proposed to minimize the average distortion caused by bit errors. For packet-erasure channels, the multiple description BSA (MD-BSA) [11] was proposed to optimize the index assignment for multiple description vector quantizers (MDVQ). The main disadvantage of the MD-BSA is its computational inefficiency due to the exhaustive switching operation of the quantizer codebook. This problem becomes serious especially when the quantizer codebook size is large. In order to reduce its computational complexity, we formulated the index assignment problem based on a linear programming framework and then proposed a new local search algorithm, called the multiple description Hungarian algorithm (MD-HA), to quickly find the optimal index assignment.

This thesis is organized as follows. Chapter 2 describes how to use the ns2 to simulate the packet loss characteristics of the MANET. Chapter 3 presents the MDVQ transmission scheme and the corresponding formulation of the optimal decoder design. The Hungarian algorithm for the bipartite matching problem and the proposed MD-HA are introduced in Chapter 4. In Chapter 5, experiments are conducted to investigate the performances of various index assignment algorithms and transmission schemes.

Chapter 2

NS-2 Based MANET Simulation

The concept of the MANET was originally proposed by the Packet Radio Network (PRNET) project conducted in the U.S. Defense Advance Research Projects Agency (DARPA). It was initially applied to provide robust battlefield communications. Recently, due to its flexibility, mobility, and convenient usage, this technology is adopted in more applications such as scientific data collections, disaster recovery, and crises emergency rescue where the wired communication is not available. The MANET is an autonomous system because there exists no centralized infrastructure such as fixed base stations. Every host equipped with the wireless interface (i.e. IEEE 802.11) can exchange information directly with each other if the receiver is located in the transmission range of the sender. If the destination is out of range, routing protocols will discover one optimal hop-by-hop transmission path between the communication pair. Thus, every host can also act as a router to forward packets to another destination. Due to the dynamic random movement of every host, the routing protocol must be designed to quickly respond to the changing topologies. For example, if any intermediate node of one stable path moves out of the transmission range or the link is broken due to work overload, routing protocols must react immediately to find another available link to replace the broken one. Consequently, routing protocols significantly affect the performance especially when the topologies change rapidly.

The popular routing protocols can be categorized into two different classes according to their operation methodologies. One class is proactive routing protocols (i.e. DSDV) and the other is reactive ones (i.e. AODV). Another protocol called the TORA combined these two properties. The performances of various routing protocols have been studied in [12], [13], [14], [15], and [16]. From the simulation results in terms of the packet delivery ratio, the AODV yields a better performance than the DSDV and the TORA. Therefore, we will adopt the AODV routing protocol in the following simulations.

2.1 Routing Protocols for the MANET

2.1.1 Destination-Sequenced Distance Vector (DSDV)

The DSDV [17] is typically a table-driven routing protocol. Every node establishes a routing table which records the number of hops required to reach all available nodes in the topology. A sequence number originated by the destination is tagged to each table entry. This number determines the freshness of a path. When selecting one path among two, the path which has a greater sequence number is preferable. If the two sequence numbers are equal, the path with a lower metric will be selected. Every node periodically exchanges and updates the routing table information with each other. But, when the topology changes fast, the newest routing information can not be catched and updated immediately. As a result, according to the old routing table, the data will probably be forwarded to the destination through a broken route. This condition will cause large packet loss in the presence of high mobility.

2.1.2 Ad-hoc On-Demand Distance Vector (AODV)

The AODV [18] routing protocol discovers a transmission path based on the "on-demand" fashion. Specifically, the nodes that do not act as routers or intermediate stations along a established link have no obligation to perform any path searching or maintenance operation defined in the routing protocol. The path discovery procedure will only be initiated when two nodes need to communicate. We now briefly describe the routing mechanism. As the path discovery is started, a packet called route request (RREQ) is broadcasted by the source node to its neighbors. This kind of packet will be transmitted hop-by-hop until reaching a node which has a route to the destination. Note that an intermediate node may receive multiple duplicates of the same RREQ. If this happens, the node will drop this RREQ instead of broadcasting it again. A reverse path will be automatically established when the RREQ passes through the intermediate nodes. When the RREQ arrives at a node having an available link to the destination, this node will send a packet called route reply (RREP) back to the source node along the pre-constructed reverse path. A forward pointer giving the direction where the RREP comes from will be set up when the RREP goes through every node along the reverse path. Once the source node receives the RREP, the data transmission will be started. On the other hand, if any node along the transmitting path fails to communicate, a special RREP will be broadcasted to notify every active source node. The source node can then initiate another path discovery procedure if it still need to communicate to the destination.

2.2 NS2 Simulation Platform

2.2.1 Wireless Multihop Simulation Environments

The Monarch projects, conducted at Carnegie-Mellon University, proposed a wireless multihop simulation platform which integrated the models of physical, data link, and medium access control (MAC) layers. The simulation environments are briefly described as follows. The signal transmitted in the wireless channel is simulated by two radio propagation models, including free space propagation model and two-ray ground reflection model. Lucent WaveLAN [19, 20] which has a transmission range of 250 m and a bit rate of 2 Mb/s is adopted to model the wireless interface. The IEEE 802.11 Distributed Coordination Function (DCF) which exploits physical carrier-sensing and virtual carrier-sensing implements the MAC layer protocol. Before transmitting the data, the Request-to-Send/Clear-to-Send (RTS/CTS) exchange is performed to reserve wireless channel to prevent collisions caused by hidden terminals. After receiving a packet successfully, an Acknowledge (ACK) is transmitted back to the sender.

2.2.2 Simulation Methodology

To proceed with the NS2-based MANET simulation, it is prerequisite to create a wireless scenario script which defines all the simulation settings, such as the radio-propagation model, the length of the interface buffer, the area of the simulation field, the ad-hoc routing protocol, the number of mobile nodes, etc. Another two important issues are the generation of node movements file and traffic flows file.

The node movements file records the initial positions of every mobile nodes and the information of every node's moving path. This file can be made by the following command.

```
./setdest [-n number of nodes] [-p pause time] [-s maximum speed] [-t simulation time]
[-x maximum x] [-y maximum y] > [output file]
```

This program script is located under ns/indep-utils/cmu-scen-gen/setdest directory. We now consider an example of the simulation scenario created by the following command:

```
./setdest -n 50 -p 30.0 -s 20.0 -t 900 -x 1500 -y 300 > scene-50-test
```

There are 50 mobile nodes moving in the 1500 m x 300 m simulation area. The immobility period of a node after it arrives at a predefined position is 30.0 seconds. The speed of every node is randomly selected between 0 m/s and 20.0 m/s. The total simulation time is 900 seconds. The output will be directed to the output file "scene-50-test" after the execution of the program script. Note that the pause time determines the changing frequency of the topology. If this parameter is large, it means that mobile nodes remain stationary for a large portion of the simulation time. This represents that a few topology changes will happen during the simulation. Conversely, the topology changes more frequently if the pause time is small. A typical row of the output is presented as follows:

```
$ns_ at 3.000000000000 "$node_(0) setdest 86.459628716598 36.120954036579 2.469127930945"
```

With this instruction, node_(0) begins to move toward the location (86.46, 36.12) with a speed of 2.47 m/s at time 3.0 seconds.

The traffic flow file records the data stream information of every communication pair. This file can be done by the following command:

```
ns cbrgen.tcl [-type cbr or tcp] [-nn number of nodes] [-seed seed] [-mc number of connections] [-rate packet sending rate] > [output file]
```

This program script is located under ns/indep-utils/cmu-scen-gen directory. For example, suppose that we want to create a traffic flow scenario that uses a seed of 1.0 to randomly select 20 CBR connections from 50 nodes and each connection has a packet sending rate of 4.0. The following command should be used.

```
ns cbrgen.tcl -type cbr -nn 50 -seed 1.0 -mc 20 -rate 4.0 > cbr-50-test
```

The file “cbr-50-test” will record the results. The following shows a traffic pattern recorded in the output file.

```
set cbr_(0) [new Application/Traffic/CBR]
$cbr_(0) set packetSize_ 512
$cbr_(0) set interval_ 0.25
$cbr_(0) set random_ 1
$cbr_(0) set maxpkts_ 10000
$cbr_(0) attach-agent $udp_(0)
$ns_ connect $udp_(0) $null_(0)
$ns_ at 57.154239418576259 “$cbr_(0) start”
```

This traffic pattern makes udp_(0) at time 57.15 seconds start to send 512-byte packets to null_(0) every 0.25 seconds. Note that the starting time is randomly selected between 0 seconds and 180 seconds.

2.2.3 Trace File Analysis

After setting the main simulation script and creating the node movement scenario and the traffic flow patterns, ns2 integrates these three components to produce the trace file. The trace file records the detailed information of the packet sending, receiving, and transferring occurred at every mobile nodes. A typical example is shown below.

```
s 0.572581837 _48_ AGT — 7 cbr 102 [0 0 0 0] ——- [48:0 49:1 32 0] [3] 0 3
r 1.357495411 _49_ AGT — 14 cbr 122 [13a 31 11 800] ——- [47:0 49:0 98 49] [7] 3 3
```

These two lines can be roughly interpreted as follows. The node with ID of 48 sends a 102-byte packet at the time of 0.57 seconds. UID of this transmitted packet is 7. The node with ID of 49 and mac-ID of 31 receives a 122-byte packet at the time of 1.36 seconds. UID of this received packet is 14. The node which sends the packet with UID of 14 has mac-ID of 11.

The final goal of network simulations is to obtain the performance metrics, including packet loss rate, end-to-end delay, delay jitter, etc. This can be easily done by using the language, AWK, to analyze the data recorded in the trace file. AWK, which is similar to C language, can be directly executed without being compiled in advance. The main feature of AWK is that it can easily process the file which records the data in a field-by-field type. The following shows a simple AWK program which processes the trace file to obtain the packet loss rate.

```
BEGIN{
        count1 = 0;
        count2 = 0;
}
{
        action = $1;
        node = $3;
        if(action == “s” && node == “_48_”)
        {
                count1++;
        }
        if(action == “r” && node == “_49_”)
        {
                count2++;
        }
}
END{
        packet_loss_rate = (count1 - count2) / count1;
        printf(“%f”, packet_loss_rate);
}
```

The middle part of the program is executed recursively to process every row. $1 and $3 represent the first and the third field of one row data respectively. In this example, if a row data has the first field of s and the third field of _48_, count1 will be incremented by one. This counts the number of transmitted packets. Correspondingly, the number of received packets can be obtained by counting count2 which represents the number of row data that has the first field of r and the third field of _49_.

2.3 A Simulation Example

We simulate the transmission of multiframe packets of the distributed speech recognition (DSR) system over the MANET. According to the standard ETSI ES 202 212 (v.1.1.1) [21], one multiframe packet has a fixed length of 144 bytes consisting of 2 bytes of the synchronization sequence, 4 bytes of the header field, and 138 bytes of the frame packet stream. The frame packet stream contains 24 frames and each frame records the data generated by processing 10 ms of speech. Thus, one multiframe packet with 144 bytes represents 240 ms of speech, resulting in a data rate of 4.8 kbps.

The simulation environment is described as follows. There are 50 wireless mobile nodes moving with the maximum speed of 20 m/s in a simulation area whose boundary is defined as 1500 m x 300 m. One pair of nodes is selected to perform the transmission of the DSR packet. The transmitter sends the packet with 144 bytes every 0.24 seconds. In order to simulate the network congestion conditions, we select other 10, 20, and 30

CBR traffic patterns which have the packet size of 512 bytes and the sending rate of 4 packets/s to compete the transmission bandwidth with the DSR connection. The starting time of each CBR traffic pattern is randomly chosen between 0 and 180 seconds. The pause times in our simulation are 0, 30, 60, 120, 300, 600, and 900 seconds. For each pause time, the random waypoint movement model is used to generate 10 different mobility scenarios. Thus, each data point of the simulation result is an average of 10 runs with different mobility scenarios, but with identical traffic models. The simulation time equals 900 seconds. We list the simulation parameters in Table 2.1.

Table 2.1: Simulation parameters for evaluating the performances of the split VQ (SVQ) with a single description.

Number of wireless mobile node	50
Maximum moving speed of each node	20 m/s
Simulation area	1500 m x 300 m
DSR packet size	144 bytes
DSR sending interval	0.24 seconds
Number of CBR traffic pattern	10, 20, 30
CBR packet size	512 bytes
CBR sending rate	4 packets/s
Starting time of each CBR traffic pattern	Randomly chosen between 0 and 180 seconds
Pause time	0, 30, 60, 120, 300, 600, 900 seconds
Simulation time	900 seconds

Figure 2.1 shows the performances of the DSR transmission pair competed with various numbers of CBR sources. The simulation results revealed that the network congestion condition significantly affects the packet delivery fraction. For higher traffic loads, more packets will be discarded by the router buffer, resulting in lower packet delivery fractions. For higher pause times, the mobile nodes are more stationary,

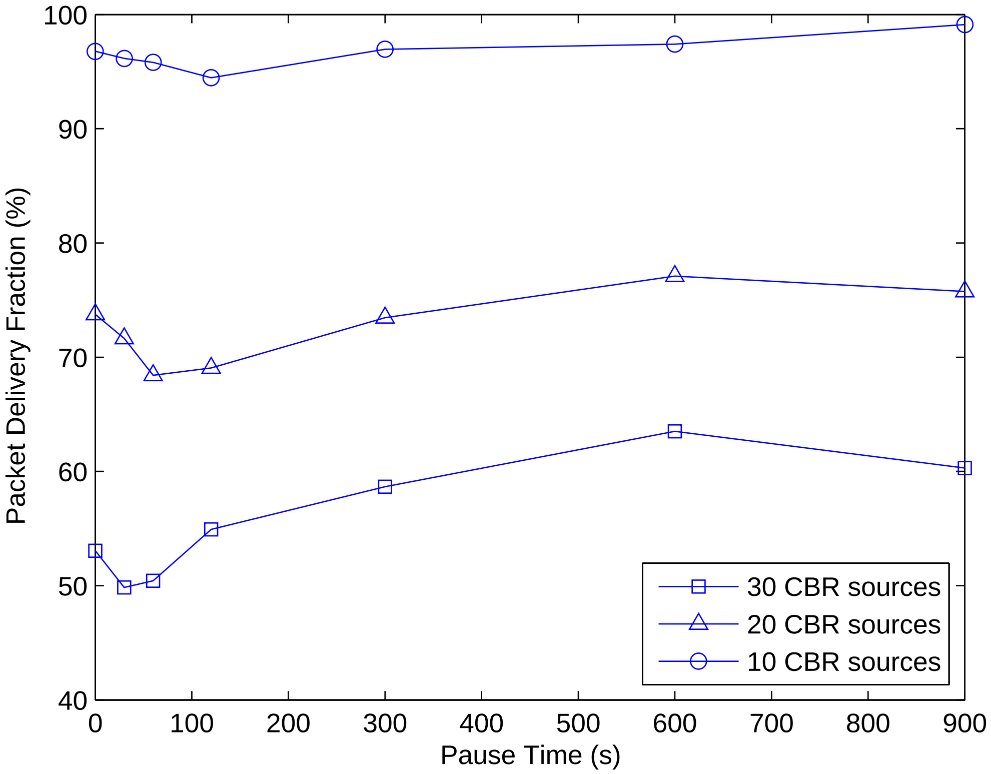


Figure 2.1: Packet delivery fractions of the DSR transmission pair competed with various numbers of CBR sources.

resulting in fewer packet losses due to the link broken caused by the topology changing. Thus, for a fixed number of CBR sources, the DSR transmission pair has slightly higher packet delivery fractions in the range of high pause times.

Chapter 3

MDVQ Transmission Scheme

The MDVQ encoder uses the fixed codebook to quantize the information source and maps the quantizer output index into two or more descriptions according to a predefined index assignments matrix. These descriptions are transmitted over multiple mutually independent and memoryless packet-erasure channels. In the decoder, given a fixed codebook and a fixed index assignment, an optimal quantizer reproduction vector is designed to minimize the expected distortion.

3.1 System Model

Figure 3.1 shows the block diagram of the MDVQ transmission scheme having two descriptions. The input source vector $\mathbf{x}$ is quantized and represented by the index of the nearest codevector $\mathbf{y}_I$ from the codebook $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_M\}$ of the VQ encoder. Let the set of quantizer output indices be denoted as $S = \{1, 2, \ldots, M\}$. The index assignments, δ_1 and δ_2, further translate the quantizer output index I into two descriptions, I_1 and I_2, whose realizations i_1, $i_2 \in \{1, 2, \ldots, N\}$. These two descriptions are then transmitted over two mutually independent and memoryless channels which have packet erasure rates ϵ_1 and ϵ_2, respectively. In the MDVQ decoder, an optimal

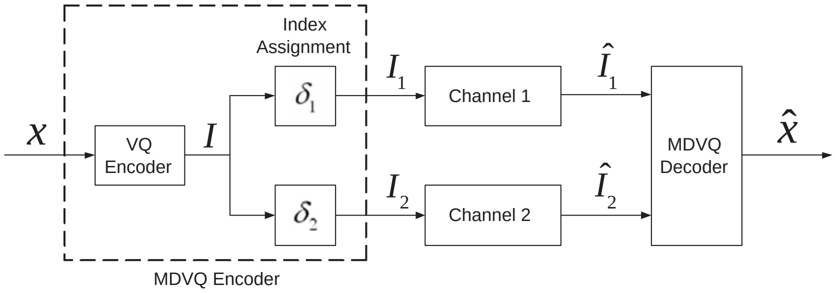


Figure 3.1: MDVQ transmission scheme.

reproduction vector $\hat{\mathbf{x}}$ is decoded based on the two received descriptions, $\hat{I}_1$ and $\hat{I}_2$, whose realization $\hat{i}_1$ and $\hat{i}_2$ take values from the set $S_1 = \{i_1, \emptyset\}$ and $S_2 = \{i_2, \emptyset\}$, respectively. Note that the empty set "$\emptyset$" indicates the case of packet erasure.

3.2 Optimal MDVQ Decoder

The optimal quantizer reproduction vector is designed to minimize the expected distortion between the codevector $\mathbf{y}_I$ and the reproduction vector $\hat{\mathbf{x}}$ for a given received pair of descriptions $\{\hat{i}_1, \hat{i}_2\}$ and a fixed codebook and a fixed index assignment. This can be described by the following optimization problem:

$$\hat{\mathbf{x}}(\hat{i}_1, \hat{i}_2) = arg \min_{\hat{\mathbf{x}}} \left\{ E \left\{ d(\mathbf{y}_I, \hat{\mathbf{x}}) | \hat{I}_1 = \hat{i}_1, \hat{I}_2 = \hat{i}_2 \right\} \right\}. \tag{3.1}$$

The distortion $d(\mathbf{y}_I, \hat{\mathbf{x}})$ will be evaluated in terms of the Euclidean distance $\|\mathbf{y_I} - \hat{\mathbf{x}}\|^2$. The minimization (3.1) leads to the optimal quantizer reproduction vector in the following form

$$\hat{\mathbf{x}}(\hat{i}_1, \hat{i}_2) = \sum_{k=1}^{M} \mathbf{y}_k P[I = k | \hat{I}_1 = \hat{i}_1, \hat{I}_2 = \hat{i}_2]. \tag{3.2}$$

The next step is to calculate the a posterior probability $P[I = k|\hat{I}_1 = \hat{i}_1, \hat{I}_2 = \hat{i}_2]$. By taking the advantage of the mutually independent property of erasure channels,

$$\begin{aligned}
&P[\hat{I}_1 = \hat{i}_1, \hat{I}_2 = \hat{i}_2|I = k] \\
&\quad = \quad P[\hat{I}_1 = \hat{i}_1, \hat{I}_2 = \hat{i}_2|I_1 = \delta_1(k), I_2 = \delta_2(k)] \\
&\quad = \quad P[\hat{I}_1 = \hat{i}_1|I_1 = \delta_1(k)] \cdot P[\hat{I}_2 = \hat{i}_2|I_2 = \delta_2(k)]. \qquad (3.3)
\end{aligned}$$

By applying the Bayes rule, we obtain

$$\begin{aligned}
&P[I = k|\hat{I}_1 = \hat{i}_1, \hat{I}_2 = \hat{i}_2] \\
&\quad = \quad \frac{P[\hat{I}_1 = \hat{i}_1|I_1 = \delta_1(k)] \cdot P[\hat{I}_2 = \hat{i}_2|I_2 = \delta_2(k)] \cdot P[I = k]}{P[\hat{I}_1 = \hat{i}_1, \hat{I}_2 = \hat{i}_2]}, \qquad (3.4)
\end{aligned}$$

where

$$P[\hat{I}_l = \hat{i}_l|I_l = \delta_l(k)] = \begin{cases} \epsilon_l, & \text{if } \hat{i}_l = \emptyset \\ 1 - \epsilon_l, & \text{if } \hat{i}_l = \delta_l(k) \end{cases} \qquad (3.5)$$

with $l = 1, 2$. Note that the sum of the left-hand side of (3.4) over all possible $k = 1, 2, \ldots, M$ equals to one. This results in

$$\begin{aligned}
&P[\hat{I}_1 = \hat{i}_1, \hat{I}_2 = \hat{i}_2] \\
&\quad = \quad \sum_{\zeta=1}^{M} P[\hat{I}_1 = \hat{i}_1|I_1 = \delta_1(\zeta)] \cdot P[\hat{I}_2 = \hat{i}_2|I_2 = \delta_2(\zeta)] \cdot P[I = \zeta]. \qquad (3.6)
\end{aligned}$$

For example, let the quantizer output index $I = i$ be transmitted over a channel condition where only the first description is received, that is $\hat{i}_1 = i_1 = \delta_1(i)$ and $\hat{i}_2 = \emptyset$. From (3.4), (3.5), and (3.6), we have

$$\begin{aligned}
&P[I = k|\hat{I}_1 = i_1, \hat{I}_2 = \emptyset] \\
&\quad = \quad \frac{P[\hat{I}_1 = i_1|I_1 = \delta_1(k)] \cdot \epsilon_2 \cdot P[I = k]}{P[\hat{I}_1 = \hat{i}_1, \hat{I}_2 = \hat{i}_2]} \qquad (3.7)
\end{aligned}$$

with

$$
\begin{aligned}
&P[\hat{I}_1 = \hat{i}_1, \hat{I}_2 = \hat{i}_2] \\
&= \sum_{\zeta=1}^{M} P[\hat{I}_1 = i_1 | I_1 = \delta_1(\zeta)] \cdot \epsilon_2 \cdot P[I = \zeta] \\
&= (1 - \epsilon_1) \cdot \epsilon_2 \sum_{\forall \zeta \in S : \delta_1(\zeta) = i_1} P[I = \zeta]. \qquad (3.8)
\end{aligned}
$$

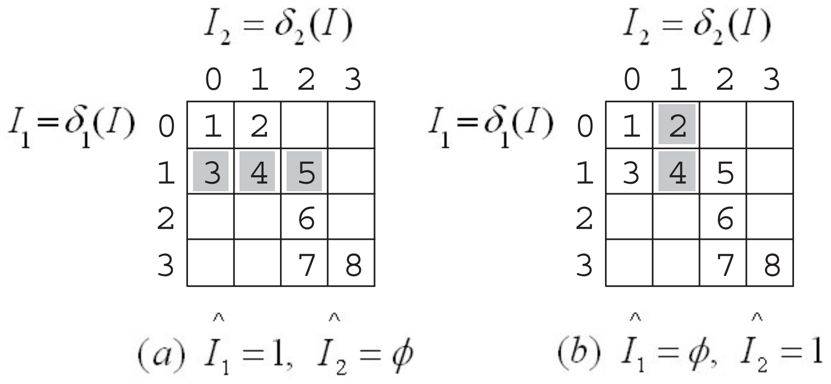
Therefore, the a posterior probability for the case where only the first description is received can be formulated as follows

$$
P[I = k | \hat{I}_1 = i_1, \hat{I}_2 = \emptyset] = \begin{cases} \frac{P[I=k]}{\sum_{\forall \zeta \in S : \delta_1(\zeta) = i_1} P[I=\zeta]}, & \text{if } \delta_1(k) = i_1 \\ 0, & \text{if } \delta_1(k) \neq i_1 \end{cases}. \qquad (3.9)
$$

Similarly, when only the second description is received, (3.10) becomes

$$
P[I = k | \hat{I}_1 = \emptyset, \hat{I}_2 = i_2] = \begin{cases} \frac{P[I=k]}{\sum_{\forall \zeta \in S : \delta_2(\zeta) = i_2} P[I=\zeta]}, & \text{if } \delta_2(k) = i_2 \\ 0, & \text{if } \delta_2(k) \neq i_2 \end{cases}. \qquad (3.10)
$$

The physical meaning of (3.9) and (3.10) is illustrated in Figure 3.2. When $\hat{I}_1 = 1$ and $\hat{I}_2 = \emptyset$, the decoding process takes the codevectors $\mathbf{y}_3$, $\mathbf{y}_4$, and $\mathbf{y}_5$ into account. On the other hand, when $\hat{I}_1 = \emptyset$ and $\hat{I}_2 = 1$, the codevectors $\mathbf{y}_2$ and $\mathbf{y}_4$ are considered.


$I_2 = \delta_2(I)$
$I_1 = \delta_1(I)$
$(a)\ \hat{I}_1 = 1,\ \hat{I}_2 = \phi$
$I_2 = \delta_2(I)$
$I_1 = \delta_1(I)$
$(b)\ \hat{I}_1 = \phi,\ \hat{I}_2 = 1$

	0	1	2	3
0	1	2		
1	3	4	5	
2			6	
3			7	8

	0	1	2	3
0	1	2		
1	3	4	5	
2			6	
3			7	8

Figure 3.2: The decoding process of the side decoder.

For the situation that both descriptions are received, the a posterior probability equals

$$P[I = k|\hat{I}_1 = \delta_1(i), \hat{I}_2 = \delta_2(i)] = \begin{cases} 1, & \text{if } k = i \\ 0, & \text{else.} \end{cases} \tag{3.11}$$

Contrarily, when both of them are erased, we obtain $P[I = k|\hat{I}_1 = \emptyset, \hat{I}_2 = \emptyset] = P[I = k]$.

By substituting the a posterior probabilities of all four possible channel conditions into (3.2), the optimal quantizer reproduction vector can be presented as follows

$$\hat{\mathbf{x}}(\hat{i}_1, \hat{i}_2) = \begin{cases} \mathbf{y}_i, & \text{if } \hat{i}_1 = i_1,\ \hat{i}_2 = i_2 \\ \sum_{k \in R_{i_1}} \frac{\mathbf{y}_k P[k]}{\sum_{\zeta \in R_{i_1}} P[\zeta]}, & \text{if } \hat{i}_1 = i_1,\ \hat{i}_2 = \emptyset \\ \sum_{k \in C_{i_2}} \frac{\mathbf{y}_k P[k]}{\sum_{\zeta \in C_{i_2}} P[\zeta]}, & \text{if } \hat{i}_1 = \emptyset,\ \hat{i}_2 = i_2 \\ \bar{\mathbf{y}} = \sum_{k=1}^{M} \mathbf{y}_k P[k], & \text{if } \hat{i}_1 = \emptyset,\ \hat{i}_2 = \emptyset \end{cases} \tag{3.12}$$

where $R_{i_1} = \{\lambda|\delta_1(\lambda) = i_1\}$ and $C_{i_2} = \{\lambda|\delta_2(\lambda) = i_2\}$ represent the sets of quantizer output indices in row i_1 and column i_2 of the index matrix, respectively.

Chapter 4

Index Assignment Algorithms

The goal of the index assignment is to place the M quantizer output indices into the $N \times N$ index matrix in such a way that the expected distortion is minimized. A straightforward way to accomplish this is to compute the distortions of all possible index assignments and select the one which has the lowest distortion. Since M locations must be selected from $N \times N$ possible elements and there are $M!$ possible combinations of the codevector indices for each selected matrix location, there exists

$$\begin{pmatrix} M \\ N \end{pmatrix} \cdot N! = \frac{M!}{(M-N)!} \tag{4.1}$$

possible index assignments. This brute-force approach is infeasible due to the large amount of computation. For example, consider the case that a five-bit quantizer is used and the quantizer output index is mapped to two four-bit descriptions. This means that there are 2^5 codevector indices assigned in an 16×16 index matrix. The distortion of totally 15.33×10^{75} different index assignments must be computed.

The MD-BSA, which modifies the single-description BSA slightly, has been proposed to alleviate the complexity problem mentioned above. The basic idea of the MD-BSA is to switch a pair of codevectors recursively until the expected channel distortion can not be further reduced. However, this recursive switching operation is still time-consuming especially when the codebook size is large. In order to further simplify the computation, we attempt to formulate the index assignment problem on the basis of a linear programming framework [22][23]. After that, we can apply the Hungarian algorithm (HA) to propose a new index assignment algorithm, referred to as the MD-HA.

4.1 Quality Criterion

Given a fixed codebook and an optimal MDVQ decoder, the expected distortion to be minimized for the index assignment design can be described as follows

$$D \doteq E\left\{d\left(\mathbf{y}_I, \hat{\mathbf{x}}(\hat{i}_1, \hat{i}_2)\right)\right\}, \tag{4.2}$$

where the optimal quantizer reproduction vector $\hat{\mathbf{x}}(\hat{i}_1, \hat{i}_2)$ can be computed by (3.12). We may expand (4.2) as follows

$$D = \sum_{k=1}^{M} P[I = k] \sum_{\forall \hat{i}_1 \in S_1} \sum_{\forall \hat{i}_2 \in S_2} d\left(\mathbf{y}_k, \hat{\mathbf{x}}(\hat{i}_1, \hat{i}_2)\right) \cdot P[\hat{I}_1 = \hat{i}_1, \hat{I}_2 = \hat{i}_2 | I = k], \tag{4.3}$$

where $S_1 = \{i_1, \emptyset\}$ and $S_2 = \{i_2, \emptyset\}$. By considering the mutually independent property of erasure channels as in (3.3), the expected distortion can be expressed as

$$D = \sum_{k=1}^{M} C(\mathbf{y}_k), \tag{4.4}$$

where the cost function

$$\begin{aligned} C(\mathbf{y}_k) \doteq{}& P[I = k] \sum_{\forall \hat{i}_1 \in S_1} \sum_{\forall \hat{i}_2 \in S_2} d\left(\mathbf{y}_k, \hat{\mathbf{x}}(\hat{i}_1, \hat{i}_2)\right) \\ & \cdot P[\hat{I}_1 = \hat{i}_1 | I_1 = \delta_1(k)] \cdot P[\hat{I}_2 = \hat{i}_2 | I_2 = \delta_2(k)]. \end{aligned} \tag{4.5}$$

By replacing the conditional probabilities of (4.5) with (3.5), we further expand the cost function as follows

$$C(\mathbf{y}_k) = P[I = k] \cdot \{d\left(\mathbf{y}_k, \hat{\mathbf{x}}(i_1, \emptyset)\right) \cdot (1 - \epsilon_1) \cdot \epsilon_2$$

$$\begin{aligned} &+d\left(\mathbf{y}_k, \hat{\mathbf{x}}(\emptyset, i_2)\right) \cdot \epsilon_1 \cdot (1-\epsilon_2) \\ &\quad +d\left(\mathbf{y}_k, \hat{\mathbf{x}}(\emptyset, \emptyset)\right) \cdot \epsilon_1 \cdot \epsilon_2\}. \end{aligned} \tag{4.6}$$

Note that the last term $d\left(\mathbf{y}_k, \hat{\mathbf{x}}(\emptyset, \emptyset)\right)$ is a positive constant independent of the selected index assignment. Thus, suppose that the two packet erasure probabilities are equal, i.e., $\epsilon_1 = \epsilon = \epsilon_2$, (4.4) may be spilt into two parts and rewritten as follows

$$D = (1-\epsilon) \cdot \epsilon \cdot \underbrace{\sum_{k=1}^{M} \Delta C(\mathbf{y}_k)}_{\doteq \Delta D} + \epsilon^2 \underbrace{\sum_{k=1}^{M} P[I=k] \cdot d\left(\mathbf{y}_k, \hat{\mathbf{x}}(\emptyset, \emptyset)\right)}_{\text{positive constant}} \tag{4.7}$$

with

$$\Delta C(\mathbf{y}_k) = P[I=k]\{d\left(\mathbf{y}_k, \hat{\mathbf{x}}(i_1, \emptyset)\right) + d\left(\mathbf{y}_k, \hat{\mathbf{x}}(\emptyset, i_2)\right)\}. \tag{4.8}$$

Therefore, for the minimization of the expected distortion D, it suffices to minimize ΔD. From (4.7) and (4.8), we found that ΔD is independent of the packet erasure probability ϵ. This means the resultant index assignment is fixed for all possible values of erasure probability. This is not true, however, when the two packet erasure probabilities are different.

4.2 Multiple Description Binary Switching Algorithm

The BSA was first proposed to optimize the single description index assignment problem with channels corrupted by bit errors. The basic idea of the BSA is to sort the codevectors in decreasing order of their cost values and select the codevector having the highest cost value to switch with a partner. The switch partner is chosen such that the reduced amount of the distortion due to the index switch is the largest. If there does not exist any such switch partner, the codevector with the second highest cost value will be picked up to switch next. This procedure repeats until a switch is found to lower the distortion. Following an accepted switch, all the cost values of the codevectors need to be recalculated and a new ordered list of codevectors is obtained for the next step. The switching algorithm continues until there exists no switching such that the distortion will be reduced further.

From Section 4.1, the distortion for the MDVQ system equals to the sum of the cost values of the codevectors. This distortion formulation is the same as the one of the single description system whose index assignment problem is solved by the standard BSA. Therefore, we can easily extend the concept of the standard BSA to the multiple description index assignment design. The distortion and the cost value of each codevector can be computed by (4.4) and (4.5), respectively. Like the standard BSA, the codevectors are sorted according to their cost values in decreasing order and the codevector which has the highest cost value is selected to switch with a partner. The major difference is that the switch partner can either be another codevector or an empty location in the index matrix. This is illustrated in Figure 4.1. The switch

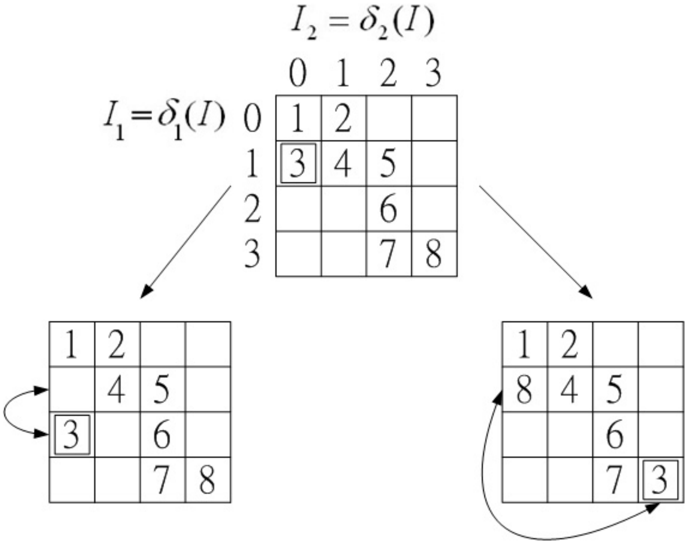


Figure 4.1: Two possible ways of switching codevectors.

partner is selected in such a way that the decrease of the distortion due to the index switch is the greatest. After an accepted switch, the distortion and the cost values of all codevectors must be recomputed and a new list of codevectors is generated for the next step. This process repeats until no possible switch can further reduce the distortion.

4.3 Bipartite Matching Problem and Hungarian Algorithm

The concept of the bipartite matching is to assign “tasks” to “individuals” subject to the constraint that each task can only be assigned to one individual and each individual is assigned to only one task. The objective is to find an assignment which leads to the minimum value of total cost. We only consider the balanced condition for which there are m individuals and m tasks. Figure 4.2 shows the structure of this problem. Each individual is associated with m cost values and so as each task. Let $x_{i,j}$ be a decision variable such that $x_{i,j} = 1$ if the task j is assigned to the individual i. The bipartite matching problem can be formulated as a linear programming problem written as follows

$$\text{minimize } z = \sum_{i=1}^{m}\sum_{j=1}^{m} c_{i,j}x_{i,j} \tag{4.9}$$

subject to:

$$\sum_{j=1}^{m} x_{i,j} = 1, \quad \text{for } i = 1, \ldots, m \tag{4.10}$$

$$\sum_{i=1}^{m} x_{i,j} = 1, \quad \text{for } j = 1, \ldots, m \tag{4.11}$$

$$x_{i,j} = 0, 1, \quad \text{for } i = 1, \ldots, m;\ j = 1, \ldots, m \tag{4.12}$$

Generally, this linear programming problem can be solved by the simplex method. However, due to the additional property of the basic feasible solution, the efficiency of

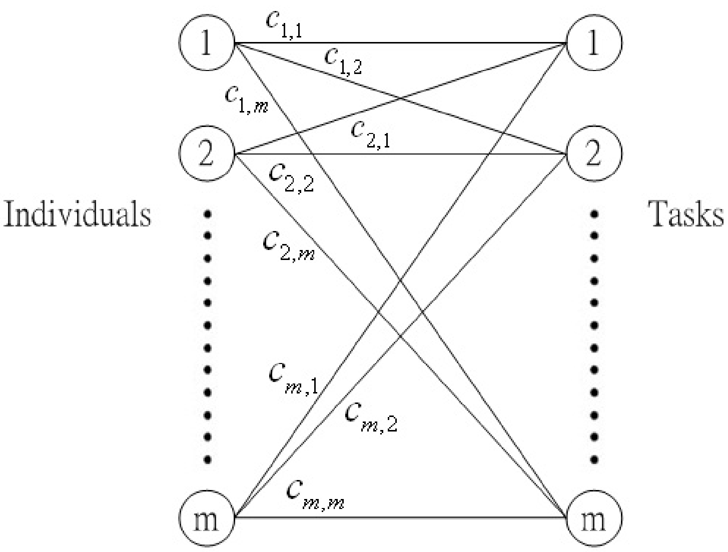


Figure 4.2: Network structure of the bipartite matching problem.

the simplex method is dramatically reduced. In order to overcome this problem, several alternative algorithms have been proposed to solve the bipartite matching problem. Among them, the Hungarian algorithm, which was developed by Kuhn in [24], finds a good solution based on the linear programming framework. Note that the linear programming problem is also referred as the primal problem.

According to the asymmetric form of duality, the dual problem of (4.9) is written as follows

$$\text{maximize} \quad \sum_{i=1}^{m} \lambda_i + \sum_{j=1}^{m} \mu_j \tag{4.13}$$

subject to:

$$\lambda_i + \mu_j \leq c_{i,j}, \quad \text{for } i = 1, \ldots, m;\ j = 1, \ldots, m \tag{4.14}$$

$$\lambda_i,\ \mu_j \text{ unrestricted} \tag{4.15}$$

where λ_i and μ_j are dual variables corresponding to constraints of $x_{i,j}$ in (4.10) and

(4.11), respectively. By applying the complementary slackness condition, we obtain

$$x_{i,j}(c_{i,j} - \lambda_i - \mu_j) = 0, \quad \text{for } i = 1, \ldots, m;\ j = 1, \ldots, m. \tag{4.16}$$

From (4.14) and (4.16), we can formulate the feasible solution of the primal problem (4.9) as follows

$$\begin{cases} x_{i,j} = 0, & \text{if } c_{i,j} - \lambda_i - \mu_j > 0 \\ x_{i,j} = 0 \text{ or } 1, & \text{if } c_{i,j} - \lambda_i - \mu_j = 0 \end{cases} \tag{4.17}$$

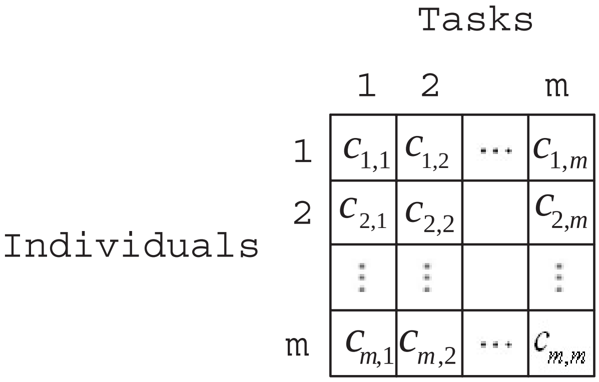
We summarize the solution progress of the bipartite matching problem as follows

1. Find feasible solutions λ_i and μ_j of the dual problem (4.13).
2. Classify the indices (i, j) into two sets: $P = \{(i, j) : c_{i,j} - \lambda_i - \mu_j > 0\}$ and $Q = \{(i, j) : c_{i,j} - \lambda_i - \mu_j = 0\}$.
3. Set $x_{i,j} = 0$, for all $(i, j) \in P$. According to the primal constraints (4.10) and (4.11), try to find a feasible solution $x_{i,j}$, for all $(i, j) \in Q$. If such a solution can be found, the solution progress terminates. Otherwise, modify the dual problem and return to Step 2.

We use the bipartite matching matrix shown in Figure 4.3 to describe the details of the Hungarian algorithm. Consider the dual solution defined as follows

$$\begin{aligned} \lambda_i &= \min_j \{c_{i,j}\} && (4.18) \\ \mu_j &= \min_i \{c_{i,j} - \lambda_i\} && (4.19) \end{aligned}$$

From (4.18), the dual variable λ_i represents the smallest element in row i of the bipartite matching matrix. For each i, all the elements in row i are subtracted by λ_i. This will generate at least one zero element in every row. We then select the dual variable μ_j as the smallest element in column j. By subtracting μ_j from every element in column j, we obtain the reduced matrix, which has entries $c_{i,j} - \lambda_i - \mu_j$.

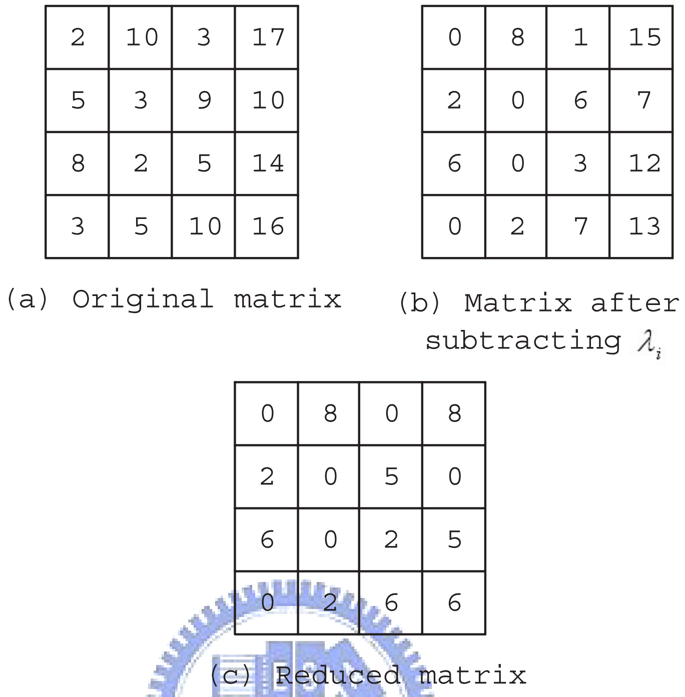


$$
\begin{array}{cc|c|c|c|c|}
 & & \multicolumn{4}{c}{\text{Tasks}} \\
 & & 1 & 2 & & m \\
\hline
 & 1 & c_{1,1} & c_{1,2} & \cdots & c_{1,m} \\
\hline
 & 2 & c_{2,1} & c_{2,2} & & c_{2,m} \\
\hline
\text{Individuals} & & \vdots & \vdots & & \vdots \\
\hline
 & m & c_{m,1} & c_{m,2} & \cdots & c_{m,m} \\
\hline
\end{array}
$$

Figure 4.3: Bipartite matching matrix.

Note that in the reduced matrix, there exists at least one zero in each row and in each column and every entry is nonnegative. Thus, due to $c_{i,j} - \lambda_i - \mu_j \geq 0$ for all (i, j), this process generates a feasible solution of the dual problem. An example of determining the reduced matrix of the bipartite matching problem with $m = 4$ is shown in Figure 4.4. From the reduced matrix, we separate the index (i, j) into two sets: $P = \{(1, 2), (1, 4), (2, 1), (2, 3), (3, 1), (3, 3), (3, 4), (4, 2), (4, 3), (4, 4)\}$ and $Q = \{(1, 1), (1, 3), (2, 2), (2, 4), (3, 2), (4, 1)\}$. According to (4.17), we set $x_{i,j} = 0$, for all $(i, j) \in P$. The remaining problem is how to decide to set $x_{i,j} = 0$ or $x_{i,j} = 1$, for $(i, j) \in Q$.

Recall from the definition of the bipartite matching: each task can only be assigned to one individual and each individual is assigned to only one task. This implies that only an assignment can be made in each row and column. Therefore, if there is only one zero in a row or column of the reduced matrix, an assignment must be made in that zero cell. We begin to find such an assignment in the reduced matrix of Figure 4.4(c). For example, as in Figure 4.5, the cell $(2, 4)$ can be selected because it is the only cell which contains the zero in column 4. We then set $x_{2,4} = 1$ and place a box around the zero in the cell $(2, 4)$. With this choice, no other assignment can be made in row 2, and therefore we cross out all other zero cells in row 2. This process continues



2	10	3	17
5	3	9	10
8	2	5	14
3	5	10	16

(a) Original matrix

0	8	1	15
2	0	6	7
6	0	3	12
0	2	7	13

(b) Matrix after subtracting λ_i

0	8	0	8
2	0	5	0
6	0	2	5
0	2	6	6

(c) Reduced matrix

Figure 4.4: Determining the reduced matrix.

until all the zero cells are either selected or crossed out. Figure 4.5 shows a final result. If exactly m assignments have been made, we obtain the optimal solution because the primal feasibility, the dual feasibility, and the complementary slackness are all satisfied. Figure 4.5 shows a typical example where exactly 4 assignments have been made. Thus, the optimal solution of the bipartite matching problem of Figure 4.4(a) equals

$$x_{1,3} = 1$$

$$x_{2,4} = 1$$

$$x_{3,2} = 1$$

$$x_{4,1} = 1$$

However, if fewer than m assignments are made, it is impossible to find a primal feasible solution corresponding to the chosen dual solution. Thus, we need to modify the

$\xcancel{0}$	8	$\boxed{0}$	8
2	$\xcancel{0}$	5	$\boxed{0}$
6	$\boxed{0}$	2	5
$\boxed{0}$	2	6	6

Figure 4.5: An optimal bipartite matching.

dual solution to introduce at least one additional zero into the reduced matrix. Assume that only k ($k < m$) assignments were made, after all the zeros are either selected or crossed out. Figure 4.6 shows such an example with $k = 2$. Before modifying the dual solution, we need to cover all the zeros in the reduced matrix with k horizontal and/or vertical lines. This can be done by the following steps:

5	7	10
4	9	6
8	6	7

(a) Original matrix

$\boxed{0}$	2	4
$\xcancel{0}$	5	1
2	$\boxed{0}$	$\xcancel{0}$

(b) Reduced matrix

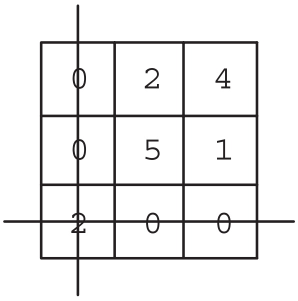
Figure 4.6: Original and reduced matrix.

1. Count the number of uncovered zeros in each row and column.

2. Draw a line through the row or column with the most uncovered zeros.

3. Repeat Steps 1 and 2 until all zeros are crossed out.

Figure 4.7 shows a final result. Next, the dual solution is modified as follows:



0	2	4
0	5	1
2	0	0

Figure 4.7: Covering all zeros with $k = 2$ lines.

1. Replace λ_i by $\lambda_i + c_0$ for all uncovered rows.

2. Replace μ_j by $\mu_j - c_o$ for all covered columns.

where c_0 is a positive constant. These two modifications have the following effects on the entries in the reduced matrix:

1. The uncovered entries decrease by c_0.

2. The entries which are covered by both a horizontal and vertical line increase by c_0.

3. The entries which are covered by a single horizontal or vertical line remain the same.

We summarize these effects in Table 4.1. The next question is how to select c_0 to introduce at least one additional zero into the reduced matrix subject to the constraint that the feasibility (i.e., $c_{i,j} - \lambda_i - \mu_j \geq 0$) of the dual solution remains unchanged. It can be seen from Table 4.1 that only the uncovered entries have the chance to become zeros. Because all of uncovered entries decrease by c_0, selecting c_0 as the minimum uncovered entry will not only result in at least one new zero into the reduced matrix, but it still also maintain the dual feasibility. The reduced matrix in Figure 4.7 indicates that $c_0 = 1$. Following the modifications summarized in Table 4.1, we have the new

Table 4.1: Effects on the entries in the reduced matrix.

Cell type	Change in $(c_{i,j} - \lambda_i - \mu_j)$
Uncovered row	$c_{i,j} - (\lambda_i + c_0) - \mu_j = (c_{i,j} - \lambda_i - \mu_j) - c_0$
Uncovered column	Therefore, $(c_{i,j} - \lambda_i - \mu_j)$ decreases by c_0
Covered row	$c_{i,j} - \lambda_i - \mu_j = c_{i,j} - \lambda_i - \mu_j$
Uncovered column	Therefore, $(c_{i,j} - \lambda_i - \mu_j)$ does not change
Uncovered row	$c_{i,j} - (\lambda_i + c_0) - (\mu_j - c_0) = c_{i,j} - \lambda_i - \mu_j$
Covered column	Therefore, $(c_{i,j} - \lambda_i - \mu_j)$ does not change
Covered row	$c_{i,j} - \lambda_i - (\mu_j - c_0) = (c_{i,j} - \lambda_i - \mu_j) + c_0$
Covered column	Therefore, $(c_{i,j} - \lambda_i - \mu_j)$ increases by c_0

reduced matrix shown in Figure 4.8. Note that an additional zero was introduced in the cell $(2, 3)$ and thus an optimal assignment is available.

$$\begin{array}{|c|c|c|}\hline \boxed{0} & 1 & 3 \\ \hline \cancel{0} & 4 & \boxed{0} \\ \hline 3 & \boxed{0} & \cancel{0} \\ \hline \end{array}$$

Figure 4.8: Modified matrix and optimal bipartite matching.

Finally, we summarize the Hungarian algorithm in the following steps:

1. Convert the original balanced cost matrix into the reduced one.

2. Apply the complementary slackness condition to make assignments in the reduced matrix. If exactly m assignments have been made, the optimal solution is obtained and thus the algorithm terminates. However, if only k $(k < m)$ assignments were made, go to Step 3.

3. Cover all the zeros in the reduced matrix with k horizontal and/or vertical lines.

4. Select c_0 as the minimum uncovered entry. Add c_0 to all the entries which are covered by both a horizontal and vertical line. Subtract c_0 from all uncovered entries. Go to Step 2.

4.4 Multiple Description Hungarian Algorithm

From (4.4) and (4.5), the expected distortion can be expanded as follows

$$D = \epsilon_1\epsilon_2 D_0 + \epsilon_2(1-\epsilon_1)D_1(\delta_1) + \epsilon_1(1-\epsilon_2)D_2(\delta_2), \tag{4.20}$$

where

$$D_l = \begin{cases} \sum_{\lambda=1}^{M} \|\mathbf{y}_\lambda - \bar{\mathbf{y}}\|^2 P[I=\lambda], & l=0; \\ \sum_{i_1=1}^{N} \sum_{\lambda\in R_{i_1}} \|\mathbf{y}_\lambda - \hat{\mathbf{x}}(i_1,\emptyset)\|^2 P[I=\lambda], & l=1; \\ \sum_{i_2=1}^{N} \sum_{\lambda\in C_{i_2}} \|\mathbf{y}_\lambda - \hat{\mathbf{x}}(\emptyset,i_2)\|^2 P[I=\lambda], & l=2. \end{cases} \tag{4.21}$$

Note that the first term is a positive constant. Therefore, for the minimization of the expected distortion D, it suffices to minimize the side distortion $D_l(\delta_l)$ with respect to δ_l for each individual description $l=1,2$.

We begin by formulating the optimization problem for assigning the row index δ_1. The MD-HA is a local search algorithm based on iterative permutation of indices in the index matrix H. This algorithm is performed on a column-by-column basis. Assume that column n is selected, the goal is to find an optimal assignment of the row index $\delta_1(\lambda)$ to indices $\lambda \in C_n$. Let $x_{ij}^{1,n}$ be an indicator variable that takes the value 1 if the index previously found in position (i,n) is moved to the new position (j,n), and takes the value of 0 otherwise. There is a cost $c_{ij}^{1,n}$ associated with this permutation in the form

$$c_{ij}^{1,n} = \sum_{\lambda\in R'_j} \|\mathbf{y}_\lambda - \sum_{k\in R'_j} \frac{\mathbf{y}_k P[I=k]}{\sum_{\zeta\in R'_j} P[I=\zeta]}\|^2 P[I=\lambda], \tag{4.22}$$

where R'_j is the new set of indices found in row j. A similar approach for assigning the column index δ_2 is developed on a row-by-row basis. If row n is selected, the cost of moving the index from position (n, i) to (n, j) is given by

$$c_{ij}^{2,n} = \sum_{\lambda \in C'_j} \|\mathbf{y}_\lambda - \sum_{k \in C'_j} \frac{\mathbf{y}_k P[I = k]}{\sum_{\zeta \in C'_j} P[I = \zeta]}\|^2 P[I = \lambda], \tag{4.23}$$

where C'_j is the new set of indices found in the column j. Also, let $x_{ij}^{2,n}$ be an indicator variable associated with this permutation.

Finding the best assignment for one column or one row is therefore a bipartite matching problem that can be solved by the Hungarian algorithm described in Section 4.3. The goal is to determine how all the permutations should be made in order to minimize the objective function

$$D^{l,n} = \sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij}^{l,n} x_{ij}^{l,n} \quad l = 1, 2 \tag{4.24}$$

subject to $\sum_{j=1}^{N} x_{ij}^{l,n} = 1$ and $\sum_{i=1}^{N} x_{ij}^{l,n} = 1$. We summarize the MD-HA as follows:

1. Set iteration index $m = 0$. Select an initial index matrix $\mathbf{H}^{(0)}$. Set initial distortion $D^{(0)} = \infty$, and a threshold $\alpha = 0.001$.

2. Perform pairwise swaps of the indices in column n and, for each swap, calculate the cost $c_{ij}^{1,n}$ using (4.22). Apply the Hungarian algorithm to determine the optimal set of indicator variables $\{x_{ij}^{1,n}\}$ that minimizes $D^{1,n}$ in (4.24).

3. Perform pairwise swaps of the indices in row n and, for each swap, calculate the cost $c_{ij}^{2,n}$ using (4.23). Apply the Hungarian algorithm to determine the optimal set of indicator variables $\{x_{ij}^{2,n}\}$ that minimizes $D^{2,n}$ in (4.24).

4. Repeat steps 2 and 3 for all values of $n = 1, 2, \ldots, N$. The corresponding sets of indicator variables are then used to update the index matrix $\mathbf{H}^{(m)}$.

5. Evaluate the distortion $D^{(m)}$ using (4.20). If $|(D^{(m)} - D^{(m-1)})/D^{(m)}| < \alpha$, stop; otherwise, let $m = m + 1$ and go to step 2.

Chapter 5

Experimental Results

In this study, we transformed the optimization of the index assignment to a bipartite matching problem and proposed the multiple description Hungarian algorithm (MD-HA) to quickly find the optimal solution. The next task is to investigate the effect of various index assignment algorithms for use in conjunction with MDVQs. A preliminary experiment was conducted to compare the performances of three index assignment algorithms: random initialization (MD-RA), the MD-BSA, and the proposed MD-HA. The performances of the MD-BSA and the MD-HA under channel-mismatched conditions were also examined.

For practical application, we conducted the Mandarin digital string recognition task to compare the performances of three coding schemes: the proposed MDVQ transmission scheme, the split VQ (SVQ) with a single description, and the FEC-protected SVQ (FEC-SVQ). In order to measure the effect of packet losses on the DSR performances, we considered three different IP network scenarios: random losses, Gilbert-model losses, and ns-2 based MANET simulation.

5.1 ETSI DSR Framework

The standard ETSI ES 202 212 (v.1.1.1) [21] describes the speech processing, transmission, and quality aspects of a distributed speech recognition (DSR) system. The front-end defines the feature extraction and an encoding scheme for low-bit-rate transmission to the remote speech recognition server. The feature extraction algorithm produces a 14-element vector consisting of a log-energy coefficient and 13 Mel-frequency cepstrum coefficients ranging from c_0 to c_{12}. For the cepstral analysis speech signals were sampled at 8 kHz and analyzed using a 25 ms Hamming window with 10 ms frame shift. Besides the feature extraction, a compression scheme is part of the front-end to transfer the speech parameters to a data stream with a rate of 4800 bits/s. The compression is based on a split VQ where the set of 14 parameters is split into 7 subsets with two coefficients in each. There are seven codebooks to map each feature pair to an entry of the corresponding codebook. Mel-frequency cepstrum coefficients c_1 to c_{10} are quantized with 6 bits each pair, c_{11} and c_{12} are quantized with 5 bits, and c_0 and the log-energy are quantized with 8 bits.

In this work, speaker independent Mandarin digit string recognition is considered as the task without restricting the string length. A mandarin digit string database recorded by 50 male and 50 female speakers was used in the experiments. Each speaker pronounced 10 utterances and 1-9 digits in each utterance. The speech of 90 speakers (45 male and 45 female) was used as the training data, and the speech of other 10 as test data. The number of digits included in the training and test data were 6796 and 642, respectively. The reference recognizer is based on the HTK software package from Entropic [25]. The digits were modelled as whole word Hidden Markov Models (HMMs) with 8 states per word and 64 mixtures for each state. In addition, a 3-state HMM was used to model pauses before and after the utterance and a one-state HMM was used to model pauses between digits. For recognition the 12 Mel-frequency cepstrum coefficients and log-energy plus the corresponding delta and acceleration coefficients are considered.

5.2 A Preliminary Experiment

The ETSI VQ codebook designed for the Mel-frequency cepstrum coefficients (c_{11}, c_{12}) was used. The corresponding codevector index has 5 bits and is mapped to two descriptions, each with 4 bits. This represents that 32 quantizer indices had to be placed into an index matrix $\mathbf{H}$ with 16×16 locations. The input source was based on a Mandarin digit string database recorded by 50 male and 50 female speakers. We used the SNR values measured on the decoder output as a performance metric to compare three index assignment algorithms: random initialization (MD-RA), the MD-BSA, and the proposed MD-HA. In implementing the MD-RA, 1000 different random index assignments were generated and selected was the one with the lowest expected distortion. We assumed that the descriptions were transmitted over two independent channels with equal frame erasure rates. Figure 5.1 shows the results in the presence of frame erasure with erasure rates ranging from 0.05 to 0.4. Both the MD-HA and the MD-BSA achieve a significant performance gain over the MD-RA. However, the improvement tends to decrease for higher erasure rates. A comparison of the MD-BSA and the MD-HA also revealed that the MD-BSA performs slightly better than the MD-HA. However, the better performance of the MD-BSA was achieved at the expense of higher computational complexity. To reach the same convergence threshold of $\alpha = 0.001$, the CPU time measured on a Pentium IV PC is 333.6 seconds for the MD-BSA, compared with 21.8 seconds for the MD-HA algorithm.

5.3 Channel-mismatch Problem

The next step in the present investigation concerned the performance degradation that may result from using the index assignment algorithms under channel-mismatch conditions. The channel-mismatch means that the packet erasure rates used for finding the optimal index assignment differ from the true erasure rates of the channels. The

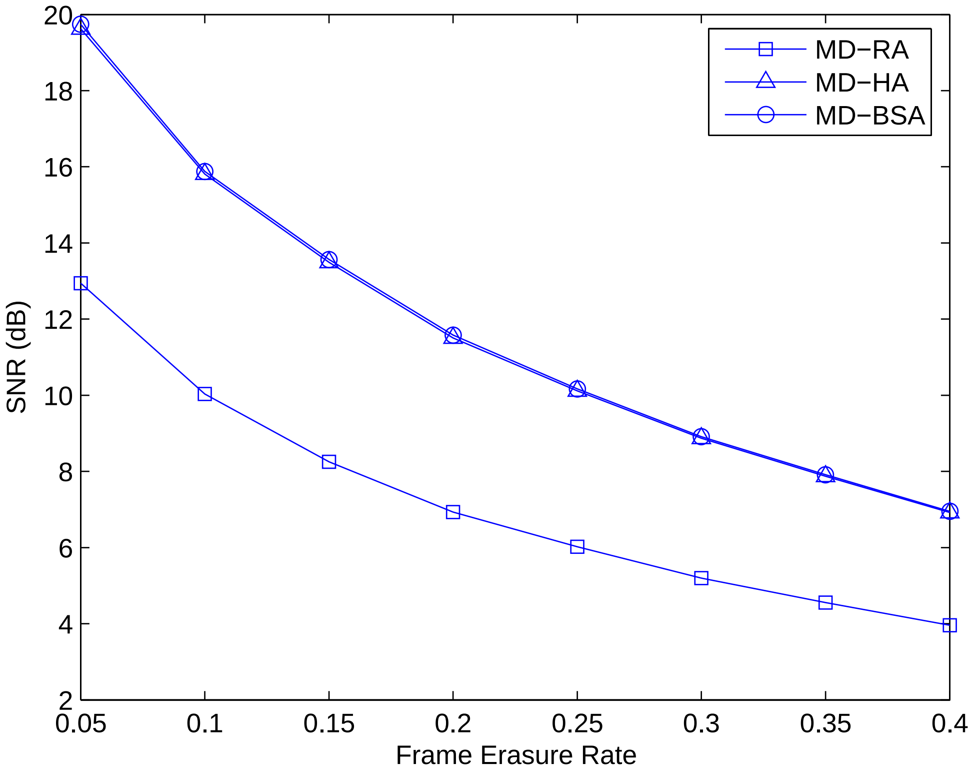


Figure 5.1: SNR performances of MDVQ with various index assignment algorithms.

effect of this problem was investigated for seven test conditions summarized in Table 5.1. The packet loss rates p_1 and p_2 of each condition were used for finding the optimal index assignment. For MDVQ transmission, we assume two channels have the true packet erasure rate of 0.3. Thus, the channel-mismatch happens in condition 1, 2, 3, 5, 6, and 7. Figure 5.2 shows the performances of the MD-BSA and the MD-HA for different test conditions. The results show that the performance of the MD-BSA degrades under channel-mismatch conditions, and the degradation tends to increase for more serious channel-mismatch conditions. On the other hand, the MD-HA achieves the same SNR in all test conditions, and even outperforms the MD-BSA in condition 1.

Table 5.1: Various test conditions.

condition	1	2	3	4	5	6	7
p_1	0.15	0.2	0.25	0.3	0.3	0.3	0.3
p_2	0.3	0.3	0.3	0.3	0.35	0.4	0.45

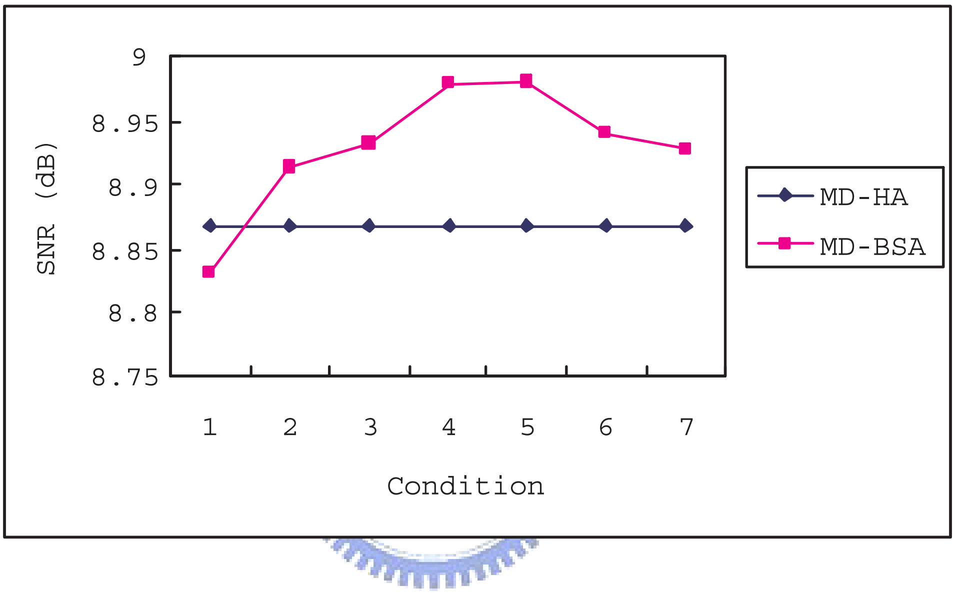


Figure 5.2: SNR performances of MD-BSA and MD-HA for different test conditions.

5.4 DSR over Symmetric Channels with Random Packet Losses

The digital recognition rate was used to compare the performances of three coding schemes: the proposed MDVQ transmission scheme, the split VQ (SVQ) with a single description, and the FEC-protected SVQ (FEC-SVQ). According to the standard ETSI 202 212 (v.1.1.1) [21], the SVQ was implemented by mapping each feature pair to a codevector index of the corresponding codebook and by packing these codevector indices into one data frame. When using the MDVQ, each data frame is split into two by mapping each codevector index to two descriptions. According to Table 5.2, the total number of bits allocated per frame is 43 and 58 for the SVQ and the MDVQ, respectively. The redundancy imposed by the MDVQ is $58/43 = 1.3$. For fair comparison, we also implemented the FEC-SVQ by exploiting the $(12, 9)$ Reed-Solomon code [26]. Figure 5.3 shows the performances of three transmission schemes in the presence of random packet losses. According to the ETSI bit-streaming format, we chose 24 frames per packet. The results in Figure 5.3 show that the proposed MDVQ outperforms other single description transmission schemes and its performance gain tends to increase for higher packet erasure rates. At the same packet erasure rate of 40%, the proposed MDVQ scheme yielded the highest recognition accuracy of 67.23 %, compared with 51.71 % and 45.23 % for the FEC-SVQ and the SVQ, respectively.

Table 5.2: DSR feature pairs and associated bit allocation.

DSR feature pair	number of bits	
	SVQ	MDVQ
(c_1, c_2)	6	$4 + 4$
(c_3, c_4)	6	$4 + 4$
(c_5, c_6)	6	$4 + 4$
(c_7, c_8)	6	$4 + 4$
(c_9, c_{10})	6	$4 + 4$
(c_{11}, c_{12})	5	$4 + 4$
(c_0, lnE)	8	$5 + 5$

5.5 DSR over Symmetric Gilbert Channels

Packet losses in the Internet usually appear in bursts and can be approximated by a 2-state Markov process known as a Gilbert model. Figure 5.4 shows this model with its transition probabilities. The state 0 and 1 represent the packet is received and lost, respectively. p is the probability that the next packet is lost given that the current one

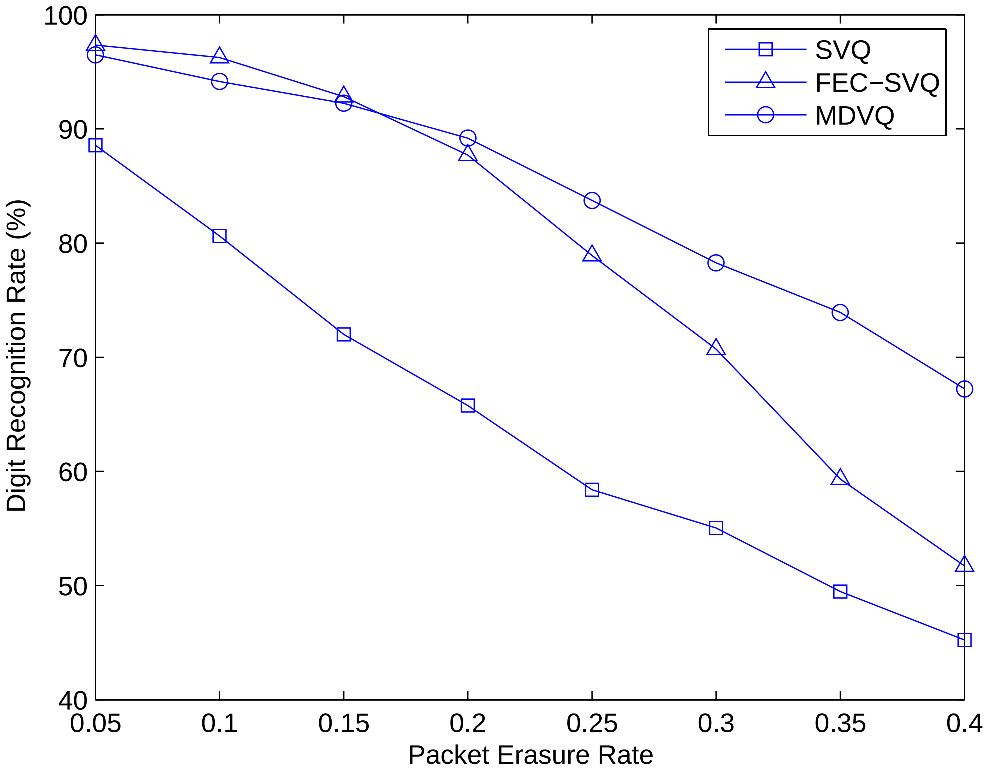


Figure 5.3: Recognition performances of various coding schemes in the presence of random packet losses.

is received; q is the probability that the next packet is received given that the current one is lost. The packet loss rate (PLR) which equals to the probability of being in the state 1 can be computed as

$$\mathrm{PLR} = p/(p+q). \tag{5.1}$$

Table 5.3 summarizes five Gilbert-model network loss conditions as described in [27]. Figure 5.5 shows the performances of three transmission schemes in the presence of Gilbert-model packet losses. The results revealed that the proposed MDVQ outperforms other single description transmission schemes and its performance gain tends

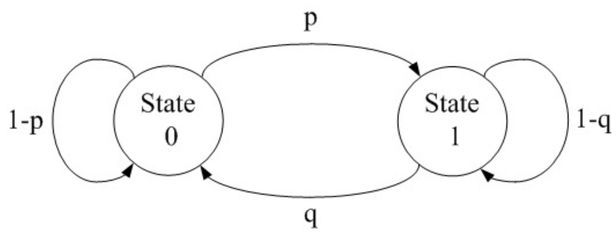


Figure 5.4: Gilbert channel model.

Table 5.3: Gilbert-model loss conditions.

condition	1	2	3	4	5
p	0.05	0.1	0.15	0.2	0.25
q	0.8	0.7	0.6	0.5	0.4
PLR	0.059	0.125	0.2	0.286	0.385

to increase for higher packet erasure rates. At the same packet erasure rate of 38.5 %, the proposed MDVQ scheme yielded the highest recognition accuracy of 75.73 %, compared with 60.19 % and 55.86 % for the FEC-SVQ and the SVQ, respectively.

5.6 DSR over Mobile Ad Hoc Networks

The MDVQ transmission scheme was applied to transmit the DSR packets over the MANET. According to the standard ETSI ES 202 212 (v.1.1.1) [21], one multiframe packet has a fixed length of 144 bytes consisting of 2 bytes of the synchronization sequence, 4 bytes of the header field, and 138 bytes of the frame packet stream. The frame packet stream contains 24 frames and 12 4-bit CRCs. Thus, according to the bit allocation of Table 5.2, one frame packet stream with 138 bytes was split to two, each with 96 bytes. The synchronization sequence with 2 bytes and the header field with 4 bytes were duplicated and added to each frame packet stream with 96 bytes, resulting in 102 bytes per multiframe packet of the MDVQ.

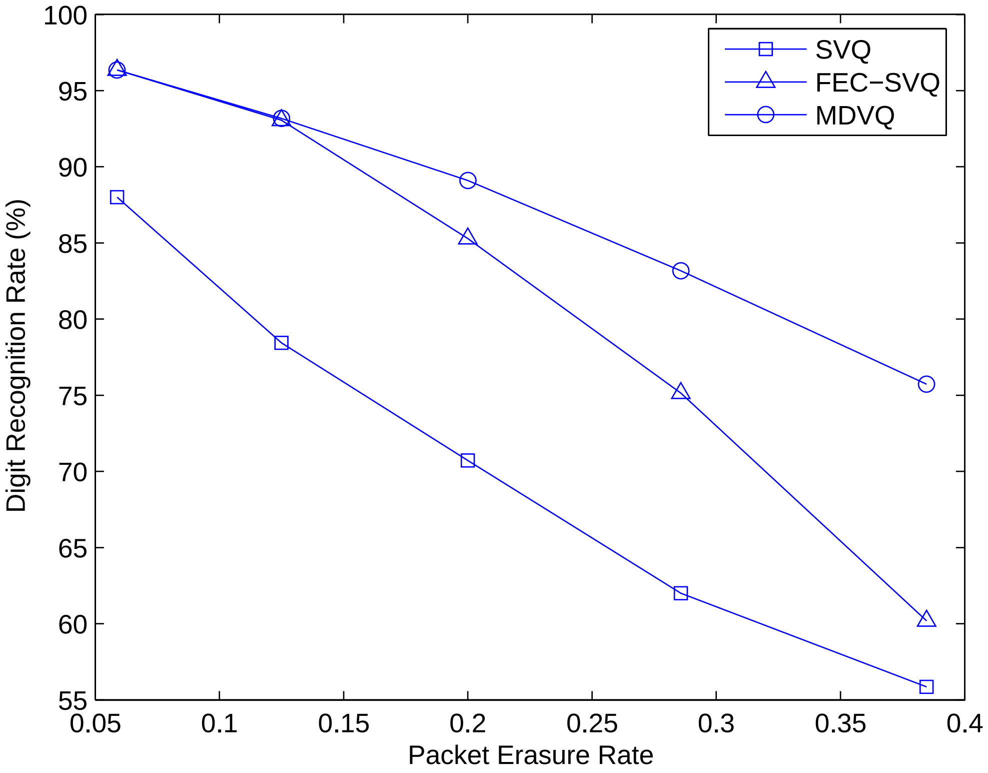


Figure 5.5: Recognition performances of various coding schemes in the presence of Gilbert-model packet losses.

The simulation environment is described as follows. There are 50 wireless mobile nodes moving with the maximum speed of 20 m/s in a simulation area whose boundary is defined as 1500 m x 300 m. Three nodes are selected to form one pair of links with the same destination to transmit the DSR packets produced by the MDVQ. Each transmitter sends the packet with 102 bytes every 0.24 seconds. In order to simulate the network congestion conditions, we select other 10, 20, and 30 CBR traffic patterns which have the packet size of 512 bytes and the sending rate of 4 packets/s to compete the transmission bandwidth with the pair of DSR connections. The starting time of each CBR traffic pattern is randomly chosen between 0 and 180 seconds. The pause times in our simulation are 0, 30, 60, 120, 300, 600, and 900 seconds. For each pause time, the random waypoint movement model is used to generate 10 different mobility scenarios. Thus, each data point of the simulation result is an average of 10 runs with different mobility scenarios, but with identical traffic models. The simulation time equals 900 seconds. We list the simulation parameters in Table 5.4.

Table 5.4: Simulation parameters for evaluating the performances of the proposed MDVQ.

Number of wireless mobile node	50
Maximum moving speed of each node	20 m/s
Simulation area	1500 m x 300 m
DSR packet size	102 bytes
DSR sending interval	0.24 seconds
Number of CBR traffic pattern	10, 20, 30
CBR packet size	512 bytes
CBR sending rate	4 packets/s
Starting time of each CBR traffic pattern	Randomly chosen between 0 and 180 seconds
Pause time	0, 30, 60, 120, 300, 600, 900 seconds
Simulation time	900 seconds

Figure 5.6 shows the performances of the proposed MDVQ using various numbers of CBR sources. The simulation results show that the network congestion condition significantly affects the digital recognition rate. For higher traffic loads, more packets will be discarded by the router buffer, resulting in lower digital recognition rates.

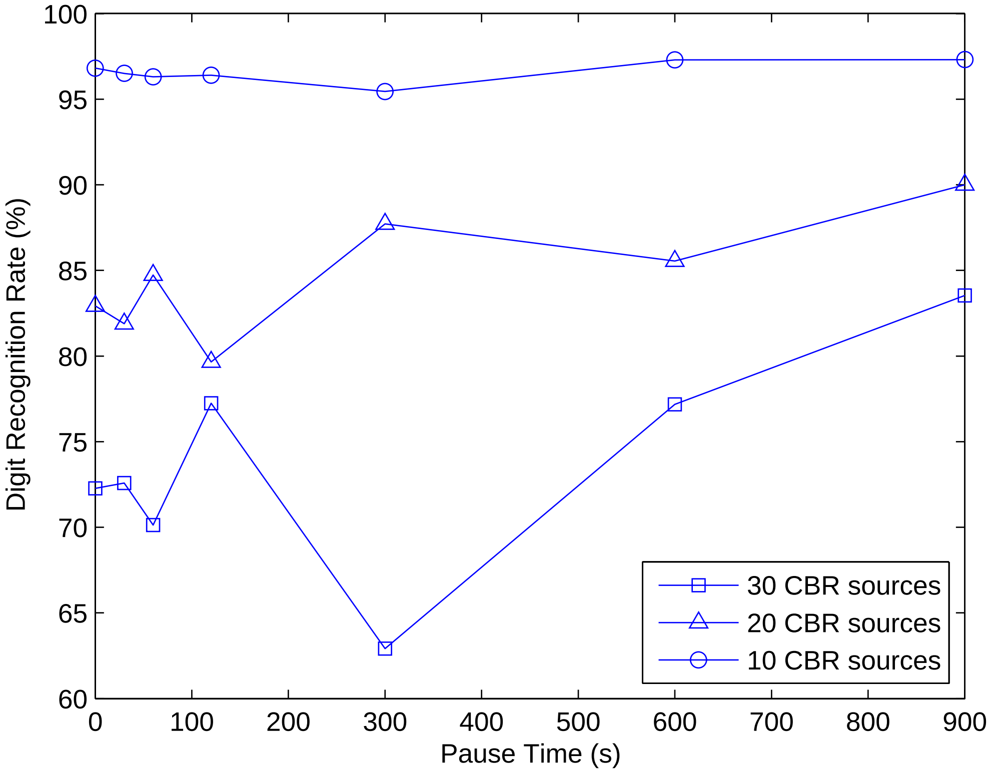


Figure 5.6: Recognition performances of the proposed MDVQ using various numbers of CBR sources.

Chapter 6

Conclusions and Future Work

In this work, we have studied the index assignment optimization for multiple description quantization and its application to robust DSR over Mobile Ad Hoc Networks. The MDVQ transmission scheme and the corresponding formulation of the optimal decoder design were first presented. The optimization of the index assignment is to place the quantizer output indices into an index matrix in such a way that the expected channel distortion is minimized. The MD-BSA accomplishes this by recursively switching a pair of codevectors until the expected channel distortion can not be further reduced. However, this exhaustive switching operation is time-consuming especially when the codebook size is large. In order to reduce its computational complexity, we formulated the index assignment problem based on a linear programming framework and then proposed a novel local search algorithm, MD-HA, to quickly find the optimal index assignment. The performance comparison between the MD-BSA and the MD-HA revealed that the slightly better performance of the MD-BSA was achieved at the expense of higher computational complexity. The performances of the MD-BSA and the MD-HA under channel-mismatched conditions were also examined.

For practical application, experiments were conducted on the Mandarin digital string recognition task under different IP network scenarios, including random losses and Gilbert-model losses. In addition, the ns-2 was introduced to simulate the packet loss characteristics of the MANET. Simulation results indicated that the proposed MDVQ achieves high robustness against random and Gilbert-model packet losses. The ns-2 based MANET simulation was also conducted to examine the DSR performances of the proposed MDVQ under Internet traffic with various numbers of CBR sources.

Finally, we present the future research issue. While, the MD-HA finds the optimal index assignment very quickly, it dose not take into consideration any channel information. In contrast, the MD-BSA uses average packet erasure probabilities of the channels to find the channel-matched index assignment. However, when a channel with memory such as the Gilbert-model channel is applied, the MD-BSA does not consider channel memory characteristics during its index assignment optimization. Thus, as a future research work, it is desired to exploit the fast search property of the MD-HA and channel memory characteristics to propose a channel-optimized index assignment algorithm.

Bibliography

[1] V. Weerackody, W. Reichl, and A. Potamianos, “An error-protected speech recognition system for wireless communications,” *IEEE Trans. Wireless Commun.*, vol. 1, no. 2, pp. 282-291, Apr. 2002.

[2] C. Boulis, M. Ostendorf, E. A. Riskin, and S. Otterson, “Graceful degradation of speech recognition performance over packet-erasure networks,” *IEEE Trans. Speech and Audio Processing*, vol. 10, no. 8, pp. 580-590, Nov. 2002.

[3] V. A. Vaishampayan, “Design of multiple description scalar quantizers,” *IEEE Trans. Inform. Theory*, vol. 39, pp. 821-834, May 1993.

[4] Y. Zhou and W.-Y. Chan, “Multiple description quantizer design using a channel optimized quantizer approach,” in *Proceedings of the 38th Annual Conference on Information Sciences and systems*, 2004.

[5] S. D. Voran, “The channel-optimized multiple-description scalar quantizer,” in *Proceedings of the 10th IEEE Digital Signal Processing Workshop*, Calloway Gardens, Pine Mountain, Georgia, Oct. 2002.

[6] P. Yahampath, “On index assignment and the design of multiple description qiantizers,” in *Proceedings of 2004 International Conference on Acoustics, Speech, and Signal Processing*, Montreal, Quebec, Canada, vol. 4, pp. iv-597-iv-600, May 2004.

[7] V. A. Vaishampayan, N. J. A. Sloane, and S. D. Servetto, “Multiple description vector quantization with lattice codebooks: Design and analysis,” *IEEE Trans. Inform. Theory*, vol. 47, pp. 1718-1734, Sep. 2001.

[8] V. K. Goyal, J. A. Kelner, and J. Kovacevic, “Multiple description vector quantization with a coarse lattice,” *IEEE Trans. Inform. Theory*, vol. 48, pp. 781-788, Mar. 2002.

[9] S. N. Diggavi, N. J. A. Sloane, and V. A. Vaishampayan, “Design of asymmetric multiple description lattice vector quantizers,” in *Proceedings of IEEE Data Compression Conference*, 2000.

[10] K. Zeger and A. Gersho, “Pseudo-Gray coding,” *IEEE Trans. Commun.*, vol. 38, no. 12, pp. 2147-2158, Dec. 1990.

[11] N. Gortz and P. Leelapornchai, “Optimization of the index assignments for multiple description vector quantizers,” *IEEE Trans. Commun.*, vol. 51, no. 3, pp. 336-340, Mar. 2003.

[12] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, “A performance comparison of multi-hop wireless ad hoc network routing protocols,” in *Proceedings of the 4th International Conference on Mobile Computing and Networking (ACM/IEEE MOBICOM ’98)*, Dallas, Texas, USA, pp. 85-97, Oct. 1998.

[13] C. E. Perkins, E. M. Royer, S. R. Das, and M. K. Marina, “Performance comparison of two on-demand routing protocols for ad hoc networks,” *IEEE Personal Communications*, vol. 8, no. 1, pp. 16-28, Feb. 2001.

[14] S. R. Das, R. Castaneda, and J. Yan, “Simulation-based performance evaluation of routing protocols for mobile ad hoc networks,” *Mobile Networks and Applications*, vol. 5, pp. 179-189, Sep. 2000.

[15] Y. Lu, Y. Zhong, and B. Bhargava, “Packet loss in mobile ad hoc networks,” Purdue University, West Lafayette, IN, 47904, Tech. Rep. CSD-TR 03-009.

[16] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, "Scenario-based performance analysis of routing protocols for mobile ad-hoc networks," in *Proceedings of the 5th International Conference on Mobile Computing and Networking (ACM/IEEE MOBICOM '99)*, Seattle, Washington, USA, pp. 195-206, Aug. 1999.

[17] C. E. Perkins and P. Bhagwat, "Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers," in *Proceedings of the SIGCOMM '94 Conference on Communications Architecture, Protocols, and Applications*, pp. 234-244, Aug. 1994. A revised version of the paper is available from http://www.cs.umd.edu/projects/mcml/papers/Sigcomm94.ps.

[18] C. E. Perkins, "Ad Hoc On Demand Distance Vector (AODV) routing," Internet draft, draft-ietf-manet-aodv-00.txt, Nov. 1997. Work in progress.

[19] D. Eckhardt and P. Steenkiste, "Measurement and analysis of the error characteristics of an in-building wireless network," in *Proceedings of the SIGCOMM '96 Conference on Applications, Technologies, Architecture, and Protocols for Computer Communication*, pp. 243-254, Aug. 1996.

[20] B. Tuch, "Development of WaveLan, an ISM band wireless LAN," *AT&T Tech. J.*, vol. 72, no. 4, pp. 27-33, Jul./Aug. 1993.

[21] ETSI ES 202 212 v1.1.1. Digital speech recognition; extended advanced front-end feature extraction algorithm; compression algorithms; back-end speech reconstruction algorithm. Nov. 2003.

[22] J. Cardinal, "Entropy-constrained index assignments for multiple description quantizers," *IEEE Trans. Signal Processing*, vol. 52, no. 1, pp. 265-270, Jan. 2004.

[23] J. P. Ignizio and T. M. Cavalier, Linear Programming, Prentice Hall, Upper Saddle River, New Jersey, 1994.

[24] H. W. Kuhn, “The Hungarian method for the assignment problem,” *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83-97, 1955.

[25] CoCentric System Studio-Reference Design Kits. Mountain View, CA: Synopsys, Inc., 2003.

[26] S. Lin and D. J. Costello, Error Control Coding, Prentice Hall, Upper Saddle River, New Jersy, Apr. 2004.

[27] H. Sanneck and N. T. L. Le, “Speech property-based FEC for Internet telephony applications,” in *Proceedings of the SPIE/ACM SIGMM Multimedia Computing and Networking Conference 2000 (MMCN 2000)*, San Jose, CA, Jan. 2000.