



142
049
THS

LIBRARY
Michigan State
University

This is to certify that the
dissertation entitled

A PROGRESSIVE RELIABILITY FRAMEWORK FOR
WIRELESS SENSOR NETWORKS

presented by

SAAD BIN QAISAR

has been accepted towards fulfillment
of the requirements for the

Ph.D. degree in Electrical Engineering


Major Professor's Signature

May 14, 2009

Date

**A PROGRESSIVE RELIABILITY
FRAMEWORK FOR WIRELESS SENSOR
NETWORKS**

By

Saad Bin Qaisar

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Electrical Engineering

2009

ABSTRACT

A PROGRESSIVE RELIABILITY FRAMEWORK FOR WIRELESS SENSOR NETWORKS

By

Saad Bin Qaisar

This dissertation investigates a new framework for achieving high data rates and negligible error probabilities by distributing the processing over multi-hop networks. In particular, we consider the case of reliable data transmission in energy constrained Wireless Sensor Networks (WSNs). Low rate channel coding can increase reliability and eliminate the need of costly retransmissions of sensor data. However, low rate channel coding on end-to-end basis puts a considerable burden in terms of transmit energy on resource constrained sensor nodes. We propose a setup that progressively provides reliability as information traverses the multi-hop wireless sensor network. Precisely, we propose an Optimal Progressive Error Recovery Algorithm (OPERA) under which, individual intermediate sensors that are relaying data toward the base station, partially and optimally channel-decode the incoming packets as data reaches the final destination. We use iteratively decodable Low Density Parity Check (LDPC)

codes in order to illustrate the efficiency of the proposed architecture. The proposed OPERA setup optimally distributes the decoding iteration budget over the entire network with minimal energy expenditure. We provide a comparison between our iteration assignment algorithm with both random iteration assignment and end-to-end channel coding, and show that OPERA performs considerably better. In addition, further motivated by resource limitation of sensor nodes and the well-known sensor reachback problem, we propose a version of OPERA that is proportionally fair to individual sensor nodes using rate adaptivity in channel coding. We use systematic puncturing of LDPC codes to develop a rate compatible framework that is fair to individual nodes by both, progressive decrease in parity as information reaches the destination node, and restricting the per node processing based on their location in the multi-hop WSN. We present various scenarios for a WSN to achieve rate-compatibility and discuss associated complexity/energy usage and distortion/reliability tradeoffs. Further motivated by distributed architectures, we propose a distributed version of OPERA in which decoding iterations are assigned in a pair-wise fashion to individual nodes. Under the proposed Distributed Progressive Error Recovery Paradigm (D-OPERA), nodes collaborate, in a distributed pair-wise manner to allocate the processing budget to individual nodes and obtain near optimal performance. We further investigate the performance of proposed framework when multiple paths are available for data originating nodes to the destination. We apply the OPERA framework to both still images as well as video streams and present an architecture for reliably transmitting video in WSNs without fast depleting their energy resources.

Copyright by
Saad Bin Qaisar
2009

Dedicated to the memory of Dr.Randy Pausch @ Carnegie Melon
University who played such a vital role in helping me understand time
management principles.

Taught man that which he knew not. (The Clot : 05)

ACKNOWLEDGMENTS

This dissertation, like any other manuscript, could not have been possible without active support of a large number of people who helped me to make it a reality. Thanks to my dissertation advisor, Dr. Hayder Radha for providing me right mix of guidance and academic freedom that helped me tremendously over the years to grow as a researcher in Electrical Engineering. His insights and suggestions have contributed greatly to this dissertation. I would also like to express my gratitude to my dissertation committee members for their valuable comments that assisted me in improving the quality of the dissertation.

Special thanks goes to my lab-mates who had been so supportive throughout my stay at WAVES Lab. To Shirish Karande with whom I had very fruitful discussions that assisted me in shaping my basic approach to problem solving. To Kiran Misra who assisted me during problem formulation. Usman Ilyas whose wits and humor adds flavor to each and every day at WAVES. And to Rami Halloush, Sohraab Soltani and 'Nima' whose presence adds value to our research group.

My family had been a constant support throughout the long and challenging phase of completing this dissertation. They provided tremendous support without which I won't be where I am, today. My father Qaisar who, with his limited resources as

a school teacher, supported and financed education for three girls and a boy who are leading successful lives today as doctors and engineers. To my mother Yasmin (Jasmine) who spent sleepless nights praying for my success, providing meals to me in my study so that I had more time to focus on next exam or next challenge in my life and instilling in me fundamental traits that helped me shape into a 'better person' each passing day. Both my parents made countless sacrifices to finance my education. They counted every single day as I had been away from them, first for five years during high school and then again, for five years during graduate school. I am fortunate to have such wonderful siblings in form of my sisters Uzma, Irum and Fariha who have always been sincere well-wishers for my success. And I won't forget to thank my surrogate mom, Dr.Colleen Vallad Hix who did not let me feel the absence of all my family members as I switched from one class to next, one paper to another.

My gratitude goes to National University of Sciences and Technology (NUST) and Higher Education Commission, Pakistan who financed my graduate studies through their fellowship programs that have enabled so many students like me to attain higher education at centers of excellence, worldwide.

Since the completion of this dissertation marks the end of twenty four years of my student life, I also owe thanks to my high school teachers who were instrumental in my success. My house masters Rashid Husseini & Abdul Qayyum and my teachers Mr.Ashfaq & Imtiaz Kassana(late) at Cadet College Hasan Abdal who worked feverishly in carving out various aspects of my personality.

Special thanks goes to Suzanne Shoemaker, a senior training specialist @ Emergent

BioDefense for teaching me believe in my abilities . To Carol Gould who had been such a great host during last one year of my dissertation writing. To my friend, Khawar Khurshid with whom I shared ups and downs of first three years of my stay in United States. And to my students whose bright faces, will to progress and ability to excel on receiving fine mentoring gave me my best moments in academia. And lastly, to my friend Sally McClintock, who, through her creation LATTICE, was instrumental in providing me a base which added so much to my insight on a land that was foreign to me yesterday and a second home, today.

TABLE OF CONTENTS

LIST OF TABLES	xiii
LIST OF FIGURES	xiv
1 Introduction	1
1.1 Processing Within the Network	2
1.1.1 Motivation for this Work	3
1.2 Overview of Contributions	3
1.3 Organization	5
2 Background and Previous Works	7
2.1 Wireless Sensor Networks	8
2.2 Binary Symmetric Channel	9
2.3 Previous Work	11
3 LDPC Statistical Model	13
3.1 LDPC Codes	14
3.2 Statistical Model	15
4 Centralized Processing Allocation	19
4.1 Problem Formulation	20
4.2 Optimal Progressive Error Recovery Algorithm	23
4.3 Fairness	25
4.3.1 Bounded OPERA ($OPERA_{bound}$)	26
4.3.2 Rate Compatible OPERA ($OPERA_{rc}$)	28
4.3.3 OPERA with Power Control	29
4.4 Simulation Setup	29
4.5 Results	31
4.5.1 End-to-end vs OPERA for a Line Network	31
4.5.2 OPERA with Multiple Flows	32
4.5.3 Random Assignment Vs OPERA	35
4.5.4 OPERA with fairness	36
4.5.5 OPERA with Power Control	41

5	Distributed Processing Allocation	43
5.1	Pair-wise D-OPERA	44
5.1.1	Pairing	46
5.2	Local Minimization Problem	47
5.2.1	Warm Start and Algorithm Convergence	48
5.3	Results	49
5.3.1	D-OPERA for a Pair	50
5.3.2	Multiple flows D-OPERA	51
6	Progressive Error Recovery with Path Diversity	54
6.1	Introduction and Motivation	55
6.2	Background	56
6.2.1	Directed Diffusion of Interest	57
6.3	Multi-Path Distributed Reliability Framework	57
6.3.1	Phase I: Data Partitioning	57
6.3.2	Phase II: Multi-Path Distributed Iteration Assignment	58
6.4	Simulation Setup	60
6.5	Results and Discussion	62
7	Application: Still Images	66
7.1	Preliminaries and Motivation	67
7.2	Simulation Setup	68
7.3	Results and Discussion	69
8	Application: Wireless Video	74
8.1	Introduction	75
8.2	Background	77
8.2.1	Video Coding Techniques	77
8.2.2	Multipath Transport (MPT)	78
8.3	Multi-Path Multi-Stream Distributed Reliability	79
8.3.1	Prioritized Data Partitioning	80
8.3.2	Path Assignment	80
8.3.3	Distributed Progressive Error Recovery	81
8.3.4	Unequal Error Protection	81
8.4	Simulation Setup	82
8.5	Results	83
9	Conclusion and Future Work	89
	APPENDICES	93
	Appendix: A	93

BIBLIOGRAPHY 100

LIST OF TABLES

8.1	H.264 Encoder Parameters	83
1	Coefficients $\alpha(\epsilon_{ind}), \beta(\epsilon_{ind}), \gamma(\epsilon_{ind}), \psi(\epsilon_{ind})$, $\epsilon_{min} = 0.01$, $\epsilon_{max} = 0.08$, $l_{max} = 150$, $R = 0.50$, $\lambda(\theta) = 0.207\theta^6 + 0.271\theta^2 + 0.522\theta^1$	94
2	Coefficients $\alpha(\epsilon_{ind}), \beta(\epsilon_{ind}), \gamma(\epsilon_{ind}), \psi(\epsilon_{ind})$, $\epsilon_{min} = 0.01$, $\epsilon_{max} = 0.08$, $l_{max} = 150$, $R = 0.51$, $\lambda(\theta) = 0.207\theta^6 + 0.271\theta^2 + 0.522\theta^1$	95
3	Coefficients $\alpha(\epsilon_{ind}), \beta(\epsilon_{ind}), \gamma(\epsilon_{ind}), \psi(\epsilon_{ind})$, $\epsilon_{min} = 0.01$, $\epsilon_{max} = 0.08$, $l_{max} = 150$, $R = 0.52$, $\lambda(\theta) = 0.207\theta^6 + 0.271\theta^2 + 0.522\theta^1$	96
4	Coefficients $\alpha(\epsilon_{ind}), \beta(\epsilon_{ind}), \gamma(\epsilon_{ind}), \psi(\epsilon_{ind})$, $\epsilon_{min} = 0.01$, $\epsilon_{max} = 0.08$, $l_{max} = 150$, $R = 0.5$, PEG (3,6) Regular LDPC	97
5	Coefficients $\alpha(\epsilon_{ind}), \beta(\epsilon_{ind}), \gamma(\epsilon_{ind}), \psi(\epsilon_{ind})$, $\epsilon_{min} = 0.005$, $\epsilon_{max} = 0.08$, $l_{max} = 150$, $R = 0.569$, $\lambda(\theta) = 0.207\theta^6 + 0.271\theta^2 + 0.522\theta^1$	98

LIST OF FIGURES

2.1	A wireless Sensor Network	9
2.2	A binary symmetric channel with error probability ϵ	10
3.1	Bit Error Rate as a function of Decoding iterations and Channel Error probability for a rate PEG (3,6) regular LDPC code with $\epsilon_{min} = 0.025$ and $\epsilon_{max} = 0.07$	16
3.2	Root Mean Square Error vs Decoding Iterations for proposed statistical model	17
4.1	A Multi-hop line network	21
4.2	Selection of Next Optimal Point $[D^1, \Gamma(\bar{l})^1, \bar{\epsilon}]$ for Dynamic Programming algorithm	24
4.3	A wireless sensor network with tree topology	27
4.4	Progressive decrease in parity for a multihop WSN	29
4.5	End-to-end vs OPERA for a Line Network	32
4.6	Optimum curves for Expected Bit Error Rate vs. Energy spent for OPERA with varying network size	33
4.7	OPERA vs Random Iteration assignment	35
4.8	$OPERA_{bound}$ with per node a) $l_{cap} = 300$ and b) $l_{cap} = 30$	37
4.9	$OPERA_{bound}$ with $l_{cap} = 30$ and $l_{cap} = 25$, $R_N = 0.50$	38
4.10	$OPERA$ vs $OPERA_{rc}$ for a) $R_{N_2} = 0.51$ and b) $R_{N_2} = 0.51, R_{N_3} = 0.52$	39
4.11	$OPERA_{brc}$ with $R_{N_2} = 0.51$, $l_{cap} = 30$, $l_{cap} = 60$ and $R_{N_2} = 0.51$, $R_{N_3} = 0.52$, $l_{cap} = 60$, $l_{cap} = 90$	40

4.12	OPERA vs 1) $OPERA_{rc}$ for a) $R_{N_2} = 0.51$ and b) $R_{N_2} = 0.51, R_{N_3} = 0.52$, 2) $OPERA_{pc}$ & $OPERA_{rcpc}$ for a) $P_{N_1} = 1mW, P_{N_2} = 0.98mW, P_{N_3} = 0.96mW$ b) $P_{N_1} = 1mW, P_{N_2} = 0.98mW, P_{N_3} = 0.96mW, R_{N_2} = 0.51, R_{N_3} = 0.52$	41
5.1	D-OPERA Pair-wise iteration assignment	46
5.2	Expected Bit Error Rate vs Energy for D-OPERA	50
5.3	Processing Budget vs Functional Evaluations of D for D-OPERA	51
5.4	Processing Budget vs Algorithmic Runs	52
5.5	D-OPERA vs OPERA for Multiple Flows	53
6.1	A multi-hop wireless sensor network with multiple paths to destination node	56
6.2	MDR vs end-to-end for $\Gamma(\bar{l})_{pairp_1} = 4, \Gamma(\bar{l})_{pairp_2} = 6$ and $\Gamma(\bar{l})_{pairp_3} = 8, \epsilon_{av} = 0.1125, 0.1175, 0.1190$	63
6.3	MDR vs end-to-end for i) $\Gamma(\bar{l})_{pairp_1} = 4, \Gamma(\bar{l})_{pairp_2} = 4, \Gamma(\bar{l})_{pairp_3} = 4$ ii) $\Gamma(\bar{l})_{pairp_1} = 4, \Gamma(\bar{l})_{pairp_2} = 6, \Gamma(\bar{l})_{pairp_3} = 8$ iii) $\Gamma(\bar{l})_{pairp_1} = 6, \Gamma(\bar{l})_{pairp_2} = 8, \Gamma(\bar{l})_{pairp_3} = 10, \epsilon_{p1} = 0.1190, \epsilon_{p2} = 0.1364, \epsilon_{p3} = 0.1190$	64
7.1	Average PSNR for D-OPERA with $\Gamma(\bar{l})_{pair} = 6$ and end-to-end for $R = 0.5$ LDPC encoded JPEG2000 image	70
7.2	JPEG2000 encoded 'boats' image with $\epsilon_{eq} = 0.14$ (clockwise from left) i- original image after source encoding ii- end-to-end channel coding iii- OPERA with $\Gamma(\bar{l})_{budget} = 6$ iv-D-OPERA with $\Gamma(\bar{l})_{pair} = 6$	71
7.3	JPEG 2000 encoded Lenna image with PERA and D-OPERA for a pair decoding budget of 6 iterations $R = 0.5$	72

7.4	JPEG2000 encoded 'lenna' image with (clockwise from left) i- original image after source encoding ii- end-to-end channel coding with $\epsilon = 0.07$ iii- PERA with $\epsilon = 0.18$ iv-D-OPERA with $\epsilon = 0.143$	73
8.1	Multi-Path Multi-Stream reliability architecture for video transmission in Wireless Sensor Networks	85
8.2	MMDR vs end-to-end transmission with $R = 0.5, K = 4, N = 4, \Gamma(\bar{l})_{pair} = 8$ a) $\epsilon_{av} = 0.12$ b) $\epsilon_{av} = 0.125$	86
8.3	Frame 2(P) for Foreman QCIF sequence	87
8.4	Average YPSNR for both MMDR with $\Gamma(\bar{l})_{pair} = 10$ and End-to-End for $R = 0.5$ LDPC encoded H.264 video.	88

CHAPTER 1

Introduction

This monograph investigates theoretical and practical aspects of data reliability in energy constrained Wireless Sensor Networks through processing within the network.

1.1 Processing Within the Network

In a multi-hop Wireless Sensor Network (WSN), reliable delivery of information to the base station is of prime importance for many applications. In many wireless sensor networks, such as machine monitoring and vehicle detection networks [4], the actual data needs to be transferred with an extremely low probability of error. End-to-end Forward Error Correction (FEC) and retransmission based schemes prove too costly for energy constrained sensor nodes [4]. In this work, we propose a more practical approach that is based on optimally distributing iterative channel decoding over sensor network. In such a paradigm, the guarantee with which the base station, or collector, gets the data from a sensor is a function of the processing within the intermediate nodes between source and destination (in-network processing [38]).

“Full processing” and “forwarding” represent two extremes of in network processing. Full processing implies complete decoding and re-encoding the information sent by the source without considering any complexity or delay constraints, thus achieving the channel capacity. Under forwarding, each intermediate node is only allowed to forward the received information without any processing thus significantly reducing the achievable data rate (assuming no path diversity). Most of the WSN applications have complexity or delay constraints and full processing may not be feasible. “Partial processing” provides an intermediate ground between forwarding and full processing.

Allowing intermediate link nodes to perform finite complexity processing achieves a significant portion of the ultimate capacity in fairly noisy networks [38].

1.1.1 Motivation for this Work

Partial recovery from errors at intermediate sensor nodes has not been studied thoroughly, and probably no work has been published in this area (to the best of our knowledge). A related context was mentioned by Fragouli et al. who discussed the benefits that can be achieved due to finite length processing at intermediate nodes [38]. Thus, motivated by achieving data reliability in sensor networks while maintaining energy efficiency, we study progressive recovery of errors in WSNs through partial decoding at intermediate nodes.

1.2 Overview of Contributions

This monograph makes several research contributions: a versatile reliability framework for Wireless Sensor Networks that provides error robustness to the data by partial processing of LDPC encoded data, a statistical model for variation of error rate with decoding iterations for LDPC codes, novel algorithms that form corner stone of processing assignment within the network: both in centralized and distributed fashion and a fairness based framework that ensures maximization in network lifetime using a novel rate compatibility structure.

We present an architecture that distributes processing over the multi-hop network such that resulting bit error rate is minimized at the destination node. In this context,

we present a centralized processing allocation algorithm (OPERA) that assigns iterations to individual nodes such that network throughput is maximized with minimal expenditure of energy within the network. We cast the problem of optimal processing within the network as rate distortion optimization and employ dynamic programming to reach the solution. We use iteratively decodable Low Density Parity Check codes to show efficiency of our scheme. We present a statistical model for variation of error rate after each LDPC decoding iteration. The model is vital for the partial processing setup where information about error rates at the end of a prescribed number of iterations is critical for optimal distribution of the iteration budget.

We not only present iteration assignment algorithm, but also fairness based framework that is proportionally fair such that sensor nodes with fewer energy resources spend lesser computational and transmit energy than those that are resource rich. Precisely, we introduce two enhancements for the centralized iteration assignment technique. A rate compatible version that uses systematic puncturing of LDPC codes to achieve fairness, and a finite processing based bounded version that puts a computational upper bound on each node in order to achieve both fairness as well as network lifetime maximization.

Though OPERA provides a progressive reliability framework, it assumes a central processor with complete topological knowledge of the network including the associated link error rates and separation between the nodes. Such global knowledge may not be available in more aggressive sensor network deployments where the network is remotely deployed, and left unattended afterwards [35]. Therefore, motivated by aforementioned facts and the centrality of OPERA, we further investigate provision of

reliability in a distributed fashion over the end-to-end path. We propose a Distributed Progressive Error Recovery Algorithm (D-OPERA) that recovers errors for a WSN in a distributed manner. Nodes collaborate, in pair-wise fashion, to distribute the progressive decoding budget. We discuss the convergence properties of the proposed scheme and ways to put least burden on the individual nodes in terms of algorithm computational overhead using warm start.

We investigate the performance of proposed framework with path diversity and selective processing budgeting based on significance of data to be transmitted over a given link. We investigate how variation in processing budget allocation to pairs of nodes affects the overall network performance at the destination node.

We consider two applications of the proposed framework: one to still images and second to video streams. Firstly, we show the performance of proposed framework when used in conjunction with still images for transmission of visual content in visual sensor networks. Secondly, we use the path-diversity framework to develop an architecture for reliable delivery of video content within a sensor network. We show the performance gains that can be achieved using the proposed framework.

1.3 Organization

The rest of the monograph is organized as follows. We formulate the problem in Chapter 2. Chapter 3 provides background on LDPC codes and relevant theorems, rate-compatible LDPC coding and presents a statistical model for variation in channel error probability with number of decoding rounds for a given LDPC code. Chapter

4 discusses the proposed approach for progressive error recovery using in-network processing in a centralized fashion and provides a fairness based architecture, whereas, we present a distributed architecture for processing assignment in Chapter 5. Chapter 6 discusses the proposed progressive error recovery framework in conjunction with path diversity. Chapter 7 and 8 apply the proposed framework to both still images and video data. We give the future directions accompanied by conclusion in Chapter 9.

CHAPTER 2

Background and Previous Works

In this chapter, we provide background to fundamental concepts that have been used throughout this monograph. Firstly, we provide a brief overview of Wireless Sensor Networks and associated research challenges.

At the end, we talk about previous works on reliability in wireless sensor networks.

2.1 Wireless Sensor Networks

Recent advances in micro-electro-mechanical systems (MEMS) technology, wireless communications and digital electronics have enabled the development of low-cost, low power, multi-functional sensor nodes that are small in size and have an ability to communicate over short distances [4]. A traditional sensor network may consist of numerous sensor nodes spread over a large geographical area. A unique feature of sensor nodes is their ability of cooperatively undertaking certain tasks. Nodes use their processing capabilities to locally carry out simple computations and transmit only the required and partially processed data.

Sensor networks enjoy a vast range of applications including those in surveillance, monitoring, intelligence, healthcare, military and product quality assurance. A more recent development is the conception of wireless multimedia and visual sensor networks that promise to add multimedia carrying capabilities to sensor nodes that have limited power, computational capacities and memory [3]. The realization of such a network necessitates overcoming various challenges including new architectures for collaborative, distributed and resource-constrained processing that allow for reliable delivery of multimedia content to the destination node. This monograph aspires to

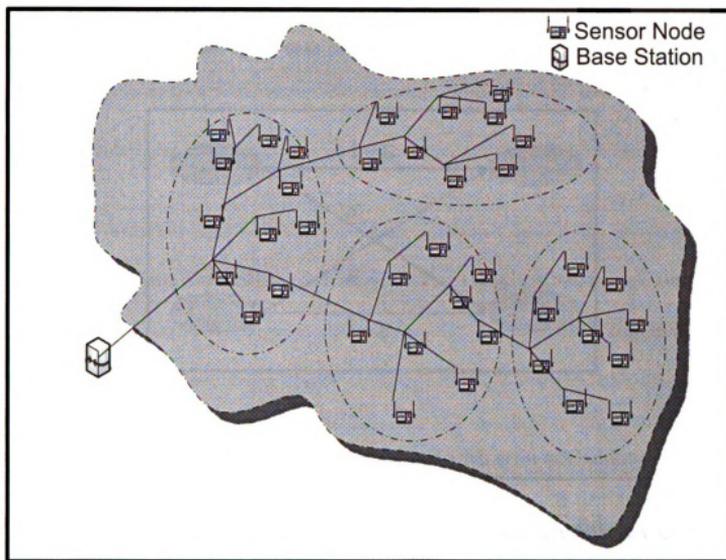


Figure 2.1. A wireless Sensor Network

overcome few of these challenges by proposing an energy efficient reliability framework for sensor networks.

2.2 Binary Symmetric Channel

A binary symmetric channel is a communications channel model used in coding and information theory. In this channel model, a transmitter intends to send a bit to the receiver and receiver wishes to receive it. In most of the cases, it is assumed that bit would be transmitted correctly, though, at times, it would be flipped with a

small 'crossover error probability' ϵ . A binary symmetric channel is used frequently in information theory due to its ease of analysis.

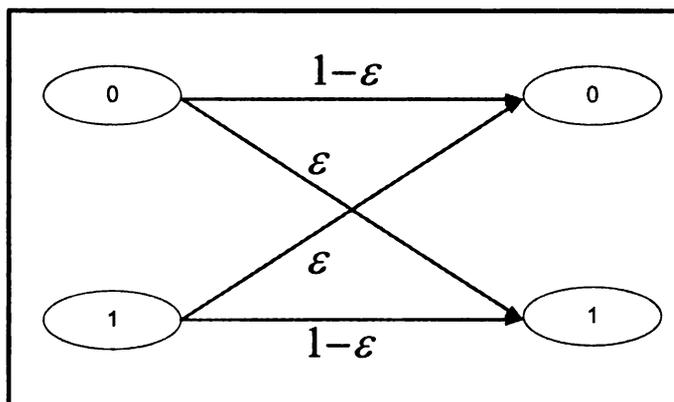


Figure 2.2. A binary symmetric channel with error probability ϵ

As shown in Figure 2.2, a BSC can transmit only one of two symbols, say a 0 or 1. A binary symmetric channel with crossover probability ϵ is a channel with binary input and binary output and error probability ϵ . If X is the transmitted random variable, and Y is the received variable, then the channel is characterized by the conditional probabilities:

$$\begin{aligned}
 Pr(Y = 0|X = 0) &= 1 - \epsilon \\
 Pr(Y = 0|X = 1) &= \epsilon \\
 Pr(Y = 1|X = 0) &= \epsilon \\
 Pr(Y = 1|X = 1) &= 1 - \epsilon
 \end{aligned} \tag{2.1}$$

where $0 \leq \epsilon \leq 1/2$.

The capacity of a binary symmetric channel is $1 - H(\epsilon)$, where $H(\epsilon)$ is the binary entropy function [8].

2.3 Previous Work

Few efforts have been made for energy efficiency based reliability in sensor networks using forward error correction. Shih et al. [35] use convolutional codes and their punctured derivatives. They conclude that for convolutional encoding and Viterbi decoding, on end-to-end coding should be used above error probability greater than as transceiver power dominates at high probabilities of error.

Sankarasubramaniam et al. [30] study the energy efficiency based packet size optimization in sensor networks and examine the effect of error control on energy efficiency. They show that some FEC coding schemes can improve the energy efficiency of a communication link, several others, including retransmissions are energy inefficient. They give a performance comparison between binary BCH codes and convolutional codes concluding that the binary BCH codes outperformed the best convolutional codes by almost 15%. Their work highlights the fact that number of FEC parities significantly impacts energy efficiency, more so than the decoding energy consumptions.

Sartipi et al. [31] presented a framework in which they consider combined source and channel coding with LDPC codes for sensor networks. A system with forward error correction (FEC) can provide an objective reliability while using less transmission power than a system without FEC. We propose to use LDPC codes for FEC. They

show that wireless sensor networks using LDPC codes are almost 45% more energy efficient than those that use BCH codes, which were shown to be 15% more energy efficient than the best performing convolutional codes.

All these works are based on end-to-end error recovery thus catering for worst case channel conditions. In terms of power consumption, transmitting an additional single bit of data is much costlier than the instructions used for computations in a sensor node [18]. For example, in a recent study, a single bit of data transmission has been shown to be equivalent to 2000 computational instructions executed in a sensor node [12]. Thus, energy tradeoff between communication and computation makes a case for processing the data inside the network rather than simply transmitting the sensor readings. Therefore, for improved reliability of data from a source node to the collector and conservation of energy, in-network processing can be highly beneficial. The proposed progressive error recovery framework ensures this by partial decoding of packets and hence enhancing the data reliability at the destination with introduction of minimal complexity as compared to end-to-end channel coding. In addition, keeping in view the energy efficiency of LDPC codes in sensor networks [31], the proposed paradigm uses them to achieve enhanced throughput while maintaining energy efficiency.

CHAPTER 3

LDPC Statistical Model

3.1 LDPC Codes

LDPC codes have gained considerable attention due to their near capacity performance. Gallager provided an algorithm for decoding of LDPC codes that is near optimal [10]. The algorithm iteratively computes the distribution of variables in graph-based models, and comes under different names/variations including Sum Product Algorithm (SPA), Belief Propagation Algorithm (BP), or more generally, Message Passing Algorithm (MPA). Richardson et al. [29] [27] give concentration theorem for LDPC codes.

Theorem 3.1.1 (LDPC Concentration Theorem) *Let $P_e^n(l)$ be the expected fraction of incorrect messages which are passed in the l th iteration of LDPC decoding, where expectation is over all instances of the code, the choice of the message and realization of the noise. For any $\delta > 0$, the probability that the actual fraction of incorrect messages which are passed in the l th iteration for any particular such instance lies outside the range converges to zero exponentially fast in n to the mean of the ensemble.*

Thus, for a sufficiently large n , we can pick up any LDPC code and the results achieved would have small deviation from the ensemble mean.

Although the concentration theorem provides a highly critical and fundamental observation regarding the performance of LDPC codes, the theorem does not provide a precise estimation model of the performance of LDPC decoding at the end of an arbitrary l th iteration. Such precise estimation is crucial for the proposed OPERA framework; in particular, when optimizing the distribution of LDPC decoding it-

erations among sensor nodes, a precise estimation for LDPC decoding model (as a function of the number of iterations) is required to provide the desired optimization. Consequently, before proceeding, we show a set of simulations that enable us to derive an estimation model for the performance of LDPC decoding as a function of an arbitrary l th decoding iteration.

For the simulations, we employ a degree seven density evolution optimized irregular, Progressive Edge Growth(PEG) based LDPC codes [15] with variable node degree polynomial $\lambda(\theta) = 0.207\theta^6 + 0.271\theta^2 + 0.522\theta^1$, message length $k = 1024$ bits. The results obtained are equally valid for any randomly picked LDPC code. We use a log-domain Sum Product Algorithm (LSPA) [10] for iterative decoding of the code which has advantages in terms of implementation, computational complexity and numerical stability [42] which is critical in the context of sensor networks.

3.2 Statistical Model

Figure 3.1 shows the expected bit error rate obtained as number of decoding iterations are increased for different channel error probabilities averaged over 1000 runs for an LDPC code, operating below code capacity. We see a sharp decrease in bit error rate as decoding iterations are increased. In fact, the decrease is exponential in nature.

Based on the above observations, we develop an exponentially decaying statistical model for the variation of bit error rate with decoding iterations. This model is then used in subsequent chapters for our proposed progressive error recovery mechanism.

Equation (3.1) expresses the relationship between the estimated bit error rate \hat{f} for

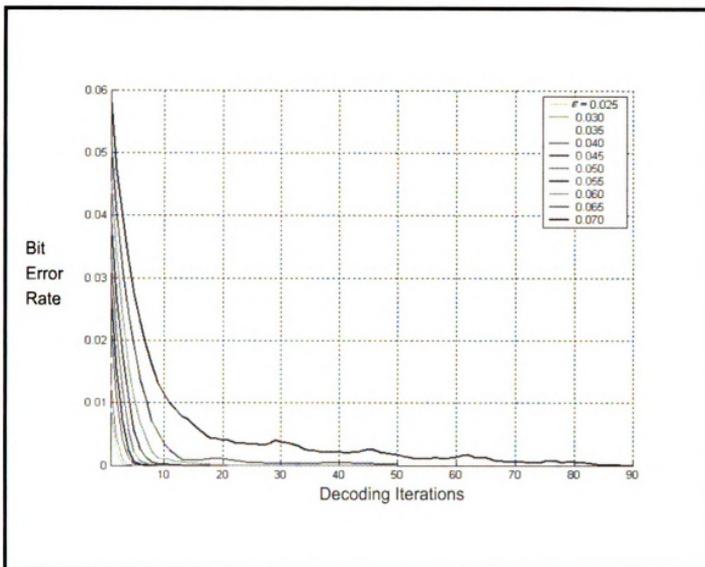


Figure 3.1. Bit Error Rate as a function of Decoding iterations and Channel Error probability for a rate PEG (3,6) regular LDPC code with $\epsilon_{min} = 0.025$ and $\epsilon_{max} = 0.07$

a given channel error probability and the number of decoding iterations for our code as

$$\hat{f}(\epsilon_{ind}, i) = \alpha(\epsilon_{ind})e^{\beta(\epsilon_{ind})i} + \gamma(\epsilon_{ind})e^{\psi(\epsilon_{ind})i} \quad (3.1)$$

Where $\epsilon_{min} \leq \epsilon \leq \epsilon_{max}$, $0 < i \leq l_{max}$, is the corresponding index value with respect to ϵ_{min} and $\alpha(\epsilon_{ind}), \beta(\epsilon_{ind}), \gamma(\epsilon_{ind})$ and $\psi(\epsilon_{ind})$ are the statistical coefficients [21].

Figure 3.2 gives the root mean square error in the proposed statistical model. It is evident from the figure that the statistical model provides a close enough approximation of the LDPC error rate variation with number of iterations.

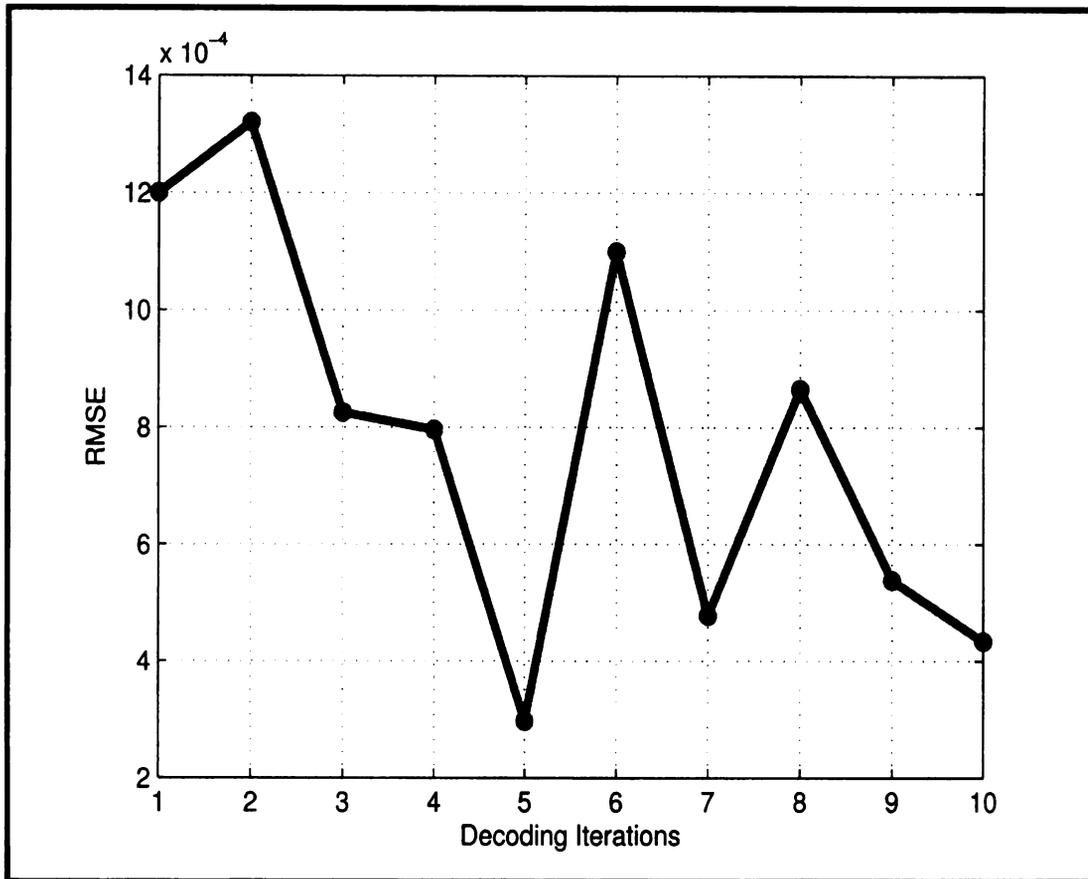


Figure 3.2. Root Mean Square Error vs Decoding Iterations for proposed statistical model

Therefore, as evident from equation (3.1) and LDPC decoding curves, significant enhancements in performance can be achieved as the number of LDPC decoding iterations are increased at the receiver provided the channel error probability is below the achievable performance bound [6] for the ensemble of codes. This result can be used to formulate an in-network processing framework to maximize the achievable



reliability at the destination.

CHAPTER 4

Centralized Processing Allocation

For the centralized processing distribution problem, we are looking for an optimal tuple $[D, \Gamma(\bar{l}), \bar{\epsilon}]$ such that $\Gamma(\bar{l}) \leq \Gamma(\bar{l})_{budget}$ i.e. a minimal overall bit error rate that meets the budget constraint. The problem can be viewed as one that allocates the overall iteration budget $\Gamma(\bar{l})_{budget}$ among all nodes in the network such that distortion is minimum.

The computational complexity of optimally mapping the iterations to the network is high, as NTM region consists of all possible operating points obtained by choosing all possible combinations of iteration assignments within the budget constraint. The hull of the NTM region would provide the desired optimal solution. We first solve the problem for a single path line network, which can then be mapped to the entire network.

4.1 Problem Formulation

A multi-hop wireless channel can be represented as a cascade of channels (Figure 4.1). Consider a line network with N nodes in cascade. We make following assumptions regarding the network:

1. The nodes in the network are quasi-stationary
2. Each node is capable of decoding the received messages
3. All nodes have equal significance and similar capabilities in processing and communication
4. Nodes are left unattended after deployment

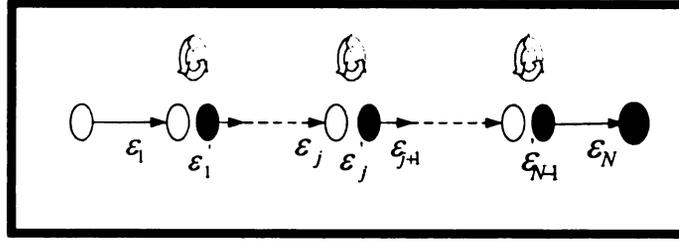


Figure 4.1. A Multi-hop line network

5. Each node has preset number of transmission power levels and at a given time, all nodes are transmitting at same power level.
6. Nodes transmit to a central coordinator over a multi-hop network

The links between the nodes are assumed to be binary symmetric with each node transmitting at power level P_T per bit. Relay nodes are allowed not only to forward the incoming information, but also to process it. In order to obtain channel error probability for a binary symmetric channel (BSC), the general expression for Signal to Interference Noise Ratio (SINR) can be given as in Equation (4.1):

$$SINR = \frac{P_T}{P_A + \sum P_{int}} \quad (4.1)$$

where P_A is the ambient noise power, and P_{int} is the interference power of any concurrent transmissions elsewhere in the network. Sources of ambient noise may include other devices operating in the same frequency band or other networks co-located with the WSN. Let T_x be the set of all the transmitting nodes in the network and $\{n_t; t \in T_x\}$ be the subset of nodes simultaneously transmitting over a certain subchannel, with transmit power P_t for each node. Then the SINR at node n_j for a transmission from node $n_i, i \in T_x$ can be calculated as in [14]:

$$SINR = \frac{\frac{P_i}{d(n_i, n_j)^\alpha}}{P_A + \sum_{t \in T, t \neq i} \frac{P_t}{d(n_t, n_j)^\alpha}} \quad (4.2)$$

Where $d(n_i, n_j)$ is the separation between n_i and n_j , and $\alpha > 2$.

If $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-u/2} du$, then the channel error probability ϵ as described in [26] is:

$$\epsilon(n_i) = Q(\sqrt{2(SINR(n_i))}) \quad (4.3)$$

The source node n_0 generates k message bits which are encoded using a rate R code.

The resulting codeword is transmitted over the first link with error probability ϵ_1 .

Node n_1 performs l_1 LDPC decoding iterations and the bit error rate in the resulting packet is a function of ϵ_1 and l_1 :

$$\epsilon_1' = f(\epsilon_1, l_1) \quad (4.4)$$

After partial processing at the second node, the error rate in the resultant packet is:

$$\epsilon_2' = f(\epsilon_2, l_2) * \epsilon_1' \quad (4.5)$$

where $\epsilon_1 * \epsilon_2 = \epsilon_2(1 - \epsilon_1) + \epsilon_1(1 - \epsilon_2)$ For the cascade line network, we define the total number of decoding iterations in the entire network as:

$$\Gamma(\vec{l}) = \sum_{j=1}^{N-1} l_j \quad (4.6)$$

where l_j is the number of decoding iterations at node n_j .

For N hops in cascade, the overall distortion measure D can be expressed as:

$$D = f(f(f(f(\epsilon_1, l_1) * \epsilon_2, l_2) * \dots * \epsilon_j, l_j) * \dots * \epsilon_{N-1}, l_{N-1}) * \epsilon_N \quad (4.7)$$

For an end-to-end path with $\bar{\epsilon} = \epsilon_1, \epsilon_2, \dots, \epsilon_j, \dots, \epsilon_{N-1}$ and given iteration budget $\Gamma(\bar{l})_{budget}$, we intend to find an iteration assignment vector $\bar{l} = l_1, l_2, \dots, l_j, \dots, l_{N-1}$ in such a way that the net throughput is maximized at the final destination. Conversely, $D(\Gamma)$ is minimized. We refer the tuple $[D, \Gamma(\bar{l}), \bar{\epsilon}]$ as Network Throughput Measure (NTM).

The problem can be seen as a budget constrained allocation problem, such that D is minimized subject to constraint $\Gamma(\bar{l}) \leq \Gamma(\bar{l})_{budget}$. From the constraint highlighted above, our problem becomes similar to minimizing a distortion measure given a budget constraint under a rate distortion framework. Therefore, an algorithm providing NTM operating point with minimum distortion D while remaining within budget constraint is desirable.

4.2 Optimal Progressive Error Recovery Algorithm

To find out the optimal hull of NTM region, we employ a dynamic programming approach similar to the method used in determination of the RD region for optimal quantizer design [33]. The algorithm stated is greedy in nature and it is possible that it may not find the optimum tuple $[D, \Gamma(\bar{l}), \bar{\epsilon}]$, though it does provide optimal solution under various practical scenarios [24] [33] [36].

The starting point of our algorithm is the case when no iterations are assigned to any intermediate nodes. Hence, D is maximum and $\Gamma(\bar{l})$ minimum. Thus the tuple

$[D^0, \Gamma(\bar{l})^0, \bar{\epsilon}]$ has D^0 closest to the D axis. The algorithm then adds a single iteration to intermediate nodes, one by one in a greedy fashion, and selects the node where the minimum D is achieved. This provides the next operating point $[D^1, \Gamma(\bar{l}^1), \bar{\epsilon}]$. The procedure is repeated till $\Gamma(\bar{l}) \leq \Gamma(\bar{l})_{budget}$ is satisfied and all the iterations are now assigned.

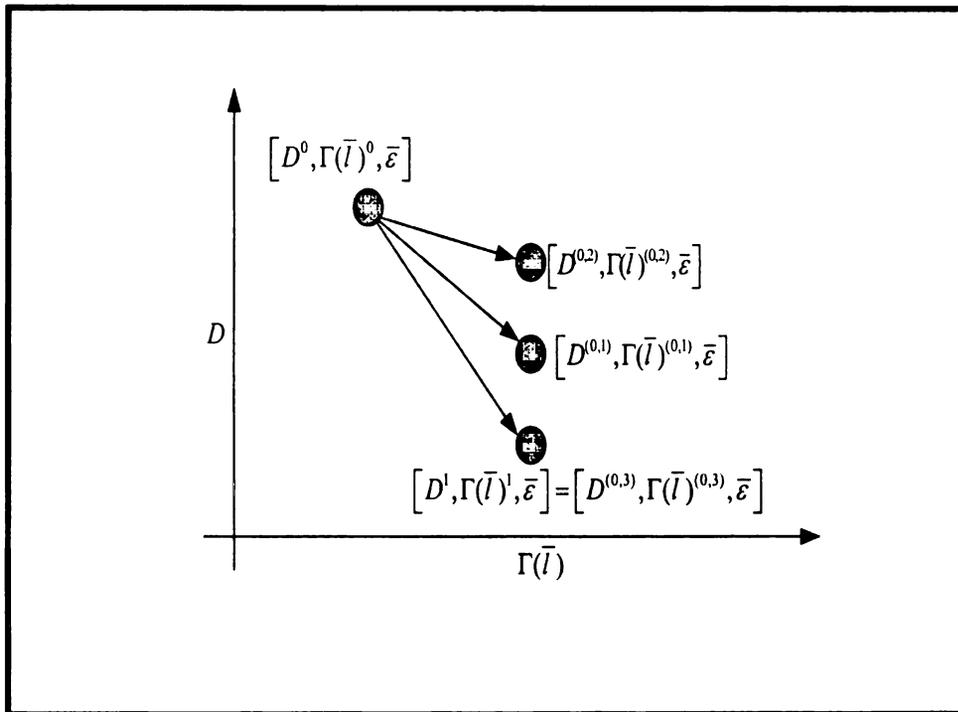


Figure 4.2. Selection of Next Optimal Point $[D^1, \Gamma(\bar{l})^1, \bar{\epsilon}]$ for Dynamic Programming algorithm

Thus, the algorithm maximizes the separation between optimal point achieved in the prior run D^- and the new points D^+ obtained in the current run for all intermediate nodes on the path. This corresponds to maximizing the gradient between the two points.

Mathematically, we have:

$$\bar{l} = \underset{j=1:N-1}{\operatorname{argmax}} \left[\frac{D^- - D^+}{\Gamma(\bar{l}^-) - \Gamma(\bar{l}_j^+)} \right] \quad (4.8)$$

where, we take $\Gamma(\bar{l}^-) - \Gamma(\bar{l}^+) = 1$.

For a multipath multihop network with nodes transmitting to a central coordinator, iterations are assigned to individual paths and the results are averaged over all the paths after decoding at base station to obtain expected bit error rate.

4.3 Fairness

Any discussion about performance of a rate control scheme must address the issue of fairness, since there exist situations where a given scheme might maximize network throughput, while denying access to some network nodes. Fairness is particularly a concern in WSN scenarios where network lifetime maximization is a prime design objective. In a typical WSN, after collecting the data of interest, the sensors must reach-back [5] and transmit the information to a single receiver. This may introduce higher complexity at few nodes closer to the receiver, than others, considering a network with multiple flows across sets of nodes that may not necessarily be disjoint. Any fairness framework for rate maximization scheme should thus consider both, reach-back and network lifetime maximization, in consideration.

Though OPERA maybe optimal in throughput sense, it may not necessarily be fair to nodes that have greater traffic loads primarily because of their closer vicinity to the base station and greater number of flows passing through them leading to the base station. In other words, we do not intend to aggravate the sensor reach-back

problem [5]. A fairness based scheme is thus greatly desired that addresses these issues. Such a scheme is particularly important in remote deployments of the network where network lifetime maximization is a prime concern. Thus, in this section, we propose a version of OPERA that is proportionally fair such that sensor nodes with lesser energy resources spend lesser computational and transmit energy than those that are resource rich. Precisely, we propose two enhancements for OPERA. A rate compatible version $OPERA_{rc}$ that uses systematic puncturing of Low Density Parity Check codes to achieve fairness, and a finite processing based bounded version $OPERA_{bound}$ that puts a computational upper bound on each node in order to achieve both fairness as well as network lifetime maximization.

4.3.1 Bounded OPERA ($OPERA_{bound}$)

We introduce an upper ceiling to the processing a node can perform for each flow passing through it based on maximum processing capacity l_{cap} of the node. Since most widely used sensor network routing algorithms are different forms of shortest path routing algorithm, we consider the routes from individual nodes to the base station forming a tree rooted at the base station [16]. We distribute the processing capacity l_{cap} based on the number of flows passing through the node. With the assumption of all the nodes transmitting simultaneously back to the base station, the number of flows passing through a node is simply the depth of sub-tree rooted at that node. Thus, greater the depth, greater the number of flows passing through the node, and lower is the number of iterations it can perform per flow. Therefore, per

4.3.2 Rate Compatible OPERA ($OPERA_{rc}$)

$OPERA_{rc}$ addresses sensor reachback problem with the introduction of network partitioning. For a multihop WSN with N nodes, we form a partition $\{N_x\}_{x \in I}$ based upon their proximity to the base station, where I belongs to set of integers. The partitions thus formed satisfy following two conditions.

$$N = \bigcup_{x \in I} N_x \quad (4.9)$$

$$N_x \cap N_y = \Phi, \{x, y\} \in I, x \neq y \quad (4.10)$$

As illustrated in Figure 4.4, we start by assigning rate R_{N1} mother code to partition $N1$. Moving further down over the end-to-end path, rate R_{N_x} punctured codes are assigned to each partition N_x till we reach the destination node. The resulting distortion over the end-to-end path is thus given as:

$$D = f_{N_{x(N-1)}}(f_{N_{x(N-2)}}(f_{N_{x(j-1)}}(f_{N_{x1}}(\epsilon_1, l_1) * \epsilon_2, l_2) * \dots \epsilon_j, l_j) * \dots \epsilon_{N-1}, l_{N-1}) * \epsilon_N \quad (4.11)$$

where, $f_{N_{xj}}(\epsilon_j, l_j)$ is the bit error rate achieved at j th node belonging to partition N_x after l_j decoding iterations with channel error probability ϵ_j . The distortion obtained is used in Equation (4.6) to get the optimal iteration assignment \bar{l} .

It is pertinent to mention here the tradeoff in design of network partitions. If the rate is increased significantly through parity drop with respect to primary rate, then the message may not remain decodable at the destination node. On the contrary, If very few bits are dropped, then we are not fair with resource constrained nodes. Thus, a balance should be maintained between performance and energy tradeoff.

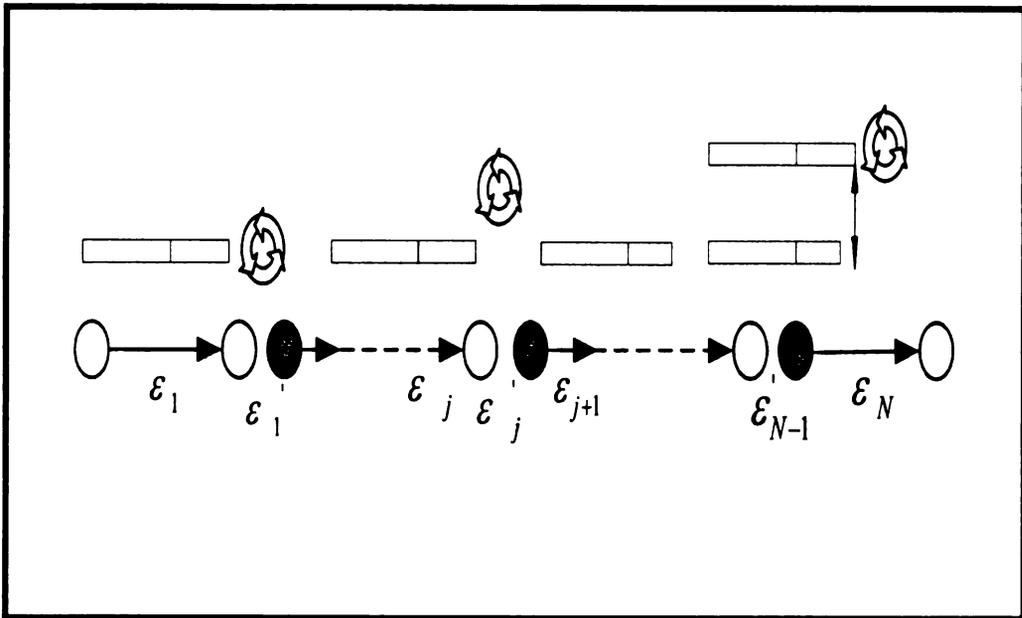


Figure 4.4. Progressive decrease in parity for a multihop WSN

4.3.3 OPERA with Power Control

$PERA_{pc}$ addresses the fairness issue by variation in transmit power for individual nodes as information reaches closer to the destination node. We use network partitioning in a similar fashion as for rate-compatible PERA. We start by assigning transmit power P_{N_1} to partition N_1 . Power levels P_{N_x} are assigned to each partition N_x as we move further down over the end-to-end path till we reach the destination node. The resulting distortion at the destination node can be given as in (11) which is used in (9) to obtain the desired iteration assignment.

4.4 Simulation Setup

For the proposed OPERA scheme, we assume that if $\epsilon < \epsilon_{min}$, only one iteration, is sufficient to completely decode the LDPC code, on average, a behavior confirmed

from the trend in Figures 3.1 and ?? and [29]. A node behaves as a forwarding node if either $\epsilon > \epsilon_{max}$ or no iteration is assigned to it. We set $\epsilon_{min} = 0.01$ and $\epsilon_{max} = 0.08$ (See [6] and [29] for tighter bound on ϵ_{max}). For a single path, we fix $\Gamma(\bar{l})_{budget} = 60$ iterations. We take per bit transmit power $P_T = 1mW$, per second, for each node unless specified otherwise (e.g. in case of power control).

The total energy spent by sensor nodes per packet delivery to the destination node is the function of computational energy and the transmission energy. We have

$$E_{total} = E_{Trans.} + E_{Comp.} \quad (4.12)$$

where, E_{comp} is the computational energy spent per packet for partial processing within the network and $E_{Trans.}$ is the per packet transmission energy for end-to-end delivery.

Fossorier et al. [9] in their work on low complexity LDPC iterative decoding, tabulate a comparison of mathematical operations required for one iteration of various LDPC decoding algorithms including MPA. Assuming each node a sensor mote equipped with Atmel Atmega128L processor, and 2000:1 ratio between per bit transmit energy and computation energy spent per instruction [12], and average number of ones per column in a parity check matrix of a code from ensemble of (n, λ, ρ) LDPC codes [32] as $\left(\int_0^1 \lambda(\theta) d\theta\right)^{-1}$ we provide performance curves for OPERA in the subsequent section using PEG irregular LDPC codes.

For a Wireless Sensor Network with multiple flows, we consider N nodes spread over a $10m \times 10m$ square grid according to a random distribution, with base station at

coordinates (5,5). Transmission range of each node is limited to $r = 2m$. Since the most widely used wireless sensor network routing algorithms (DSR, AODV, Directed Diffusion) are different forms of shortest path routing algorithms, we assume that at any given time the routes from individual nodes to the base station form a tree rooted at the base station [16]. We are thus discounting the possibility of using bifurcated routing i.e. multiple paths from source to destination. Hence, we assume that all the network nodes sending traffic back to the base station on shortest paths already established through some routing algorithm.

4.5 Results

4.5.1 End-to-end vs OPERA for a Line Network

We consider 100 instances of a line network for $N = 5$ with variation in separation between the nodes. The results achieved are averaged over all occurrences. We take end-to-end maximum and minimum rates $R_{end_{max}} = 0.731$ and $R_{end_{min}}$. For OPERA, we consider multiple rates $R_{OPERA} = 0.602, 0.569$. All the results are averaged over 1000 runs. Figure 4.5 shows the performance comparison between OPERA and end-to-end schemes with average end-to-end equivalent error probability $\epsilon_{end_{eq}}$. It is essential to note here that both OPERA and end-to-end results here are after decoding at the destination node. For end-to-end case, $E_{total} = E_{trans}$ whereas, (12) gives the total energy for OPERA. The results show that OPERA outperforms end-to-end channel decoding with much lesser energy required for a given

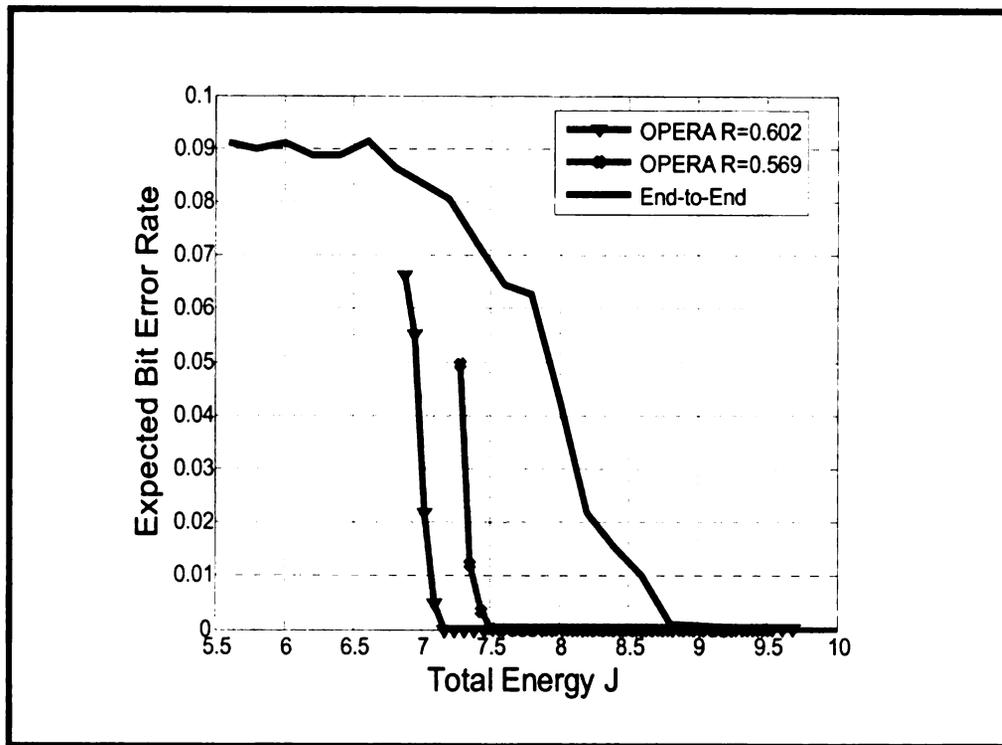


Figure 4.5. End-to-end vs OPERA for a Line Network

performance.

4.5.2 OPERA with Multiple Flows

Figure 4.6 presents the variation in throughput using OPERA for a network with multiple flows leading to the base station. The plots show that it is feasible to optimally trade-off complexity/energy usage with distortion/reliability by varying the assignment of iterations. The two extremes of the Complexity-Distortion curve in Figure 4.6. represent the performance that is offered when two extremes of in-network processing is employed. When we do not conduct any decoding at intermediate nodes, the accumulation of errors increases exponentially. In such a scenario a large number of packets eventually received at the collector may have zero or very little

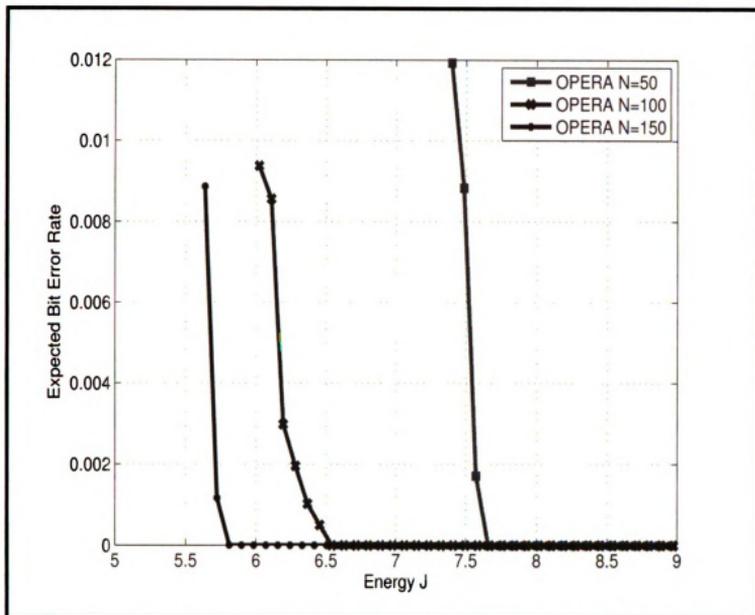


Figure 4.6. Optimum curves for Expected Bit Error Rate vs. Energy spent for OPERA with varying network size

information utility. In other extreme when decoding is employed at all or almost all the intermediate nodes, the data reliability can be increased significantly, but the energy consumption also increases. Thus, in such an operation mode, even though the throughput of a sensor network is improved, the network life-time is decreased. Our proposed approach allows us to fine-granularly operate at a large number of intermediate points. Thus in an actual deployment, the operating point can be chosen in accordance with the current demands of the network. If an important event is being sensed, we may choose to operate at high in-network processing point, as against that

if the network is in more passive state, we may prefer saving energy despite getting noisy readings.

The results in 4.6 further show that as the number of nodes increase the amount of in-network processing required to achieve improvement in reliability reduces. This can be explained by highlighting that, as the node density increases, the number of error prone links decrease. Since error recovery is primarily required only due to the presence of noisy links, decrease in the number of noisy links naturally leads to reduced requirement of network processing.

The above observation has important implications about adapting the functional usage of a sensor network through its life-time. We illustrate our point by way of an example: Let us say that a network initially composed of 150 sensors demands a functional usage represented by an error probability of 0.004. To support such a demand with in-network processing, we shall have to spend 5.70 Joules. With time as sensors die the density of sensors reduces; let's say our density drops to 100 sensors. At such point if the functional demand is not reduced the amount of energy that will have to be spent is in fact increased to 6.4 Joules. This increase in energy spending may set-off a chain reaction eventually leading to the death of a network. Thus, as the density reduces, it may be essential to adapt the functional demands from a network. The Complexity-Distortion curves obtained by our analysis can provide important guidelines on how these demands should be reduced as sensors start dieing.

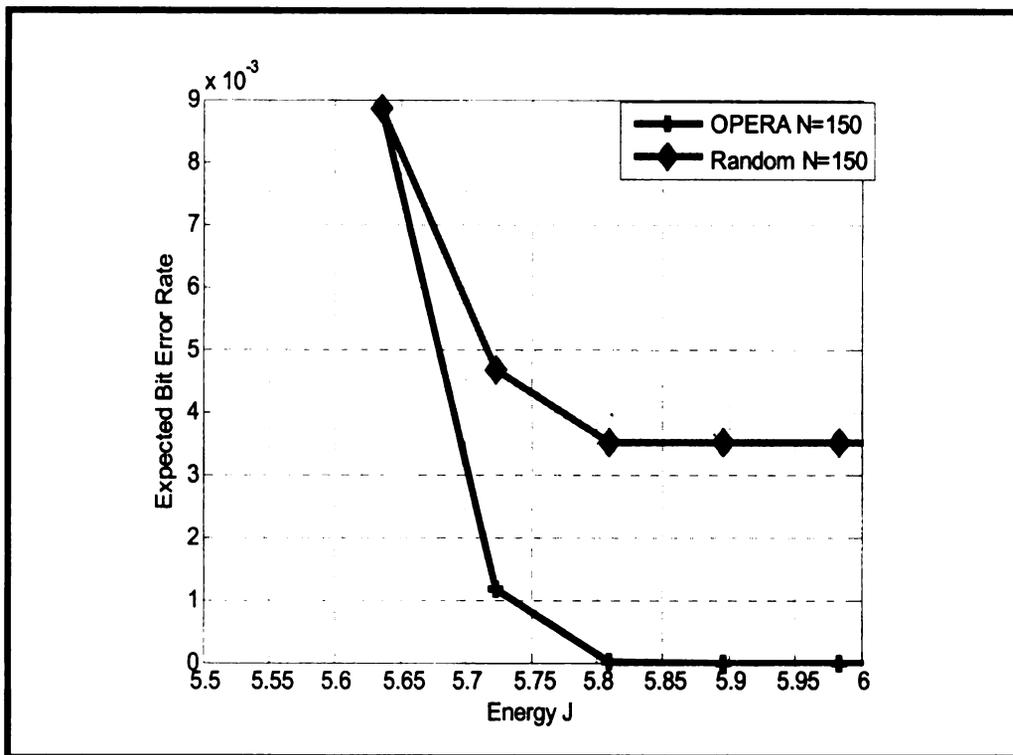


Figure 4.7. OPERA vs Random Iteration assignment

4.5.3 Random Assignment Vs OPERA

We compare the efficiency of our iteration assignment algorithm with random assignment when the decision to add iteration at a given node over the path is random such as by tossing a coin. The procedure is repeated until all the iterations are assigned to an end-to-end path. Figure 4.7 shows the efficiency of OPERA as compared to random assignment. The gap between the two curves indicates the enhancement in throughput we get by progressive iteration assignment through OPERA than randomly.

4.5.4 OPERA with fairness

Bounded OPERA

Figure 4.8 gives a comparison between iteration assignments for $OPERA_{bound}$ with $l_{cap} = 300$ and $l_{cap} = 30$, respectively, $N = 150$ and $\Gamma(\bar{l})_{budget} = 60$ iterations. In both the cases, we achieve reliable communication after destination node decoding, though, distribution of iterations is much more fair in the latter without concentration of processing load on few nodes and leaving the rest with underutilized capacity. Though, if l_{cap} is further reduced, the path budget may not get fully distributed over the end-to-end paths, thus, deteriorating performance. Figure 4.9 highlights this effect by comparing $OPERA_{bound}$ throughput curves for both $l_{cap} = 30$ and $l_{cap} = 25$. A careful selection of l_{cap} is thus significant in achieving throughput gains.

Rate Compatible OPERA

We consider two cases for $OPERA_{rc}$ with variation in network partitioning for $N = 150$. Nodes transmit at mother code rate $R_{N_1} = 0.5$ when they are either a data originating node or belong to N_1 . As the code block enters subsequent network partition, the block is subjected to rate variation as: a) $R_{N_2} = 0.5$ b) $R_{N_2} = 0.51, R_{N_3} = 0.52$.

We set $\Gamma(\bar{l})_{budget} = 60$ iterations for each path. It is assumed that each node knows the puncturing rate of the frame it would be receiving. Such information can be passed once during the initialization phase of the network, when all nodes maybe communicating at same rate.

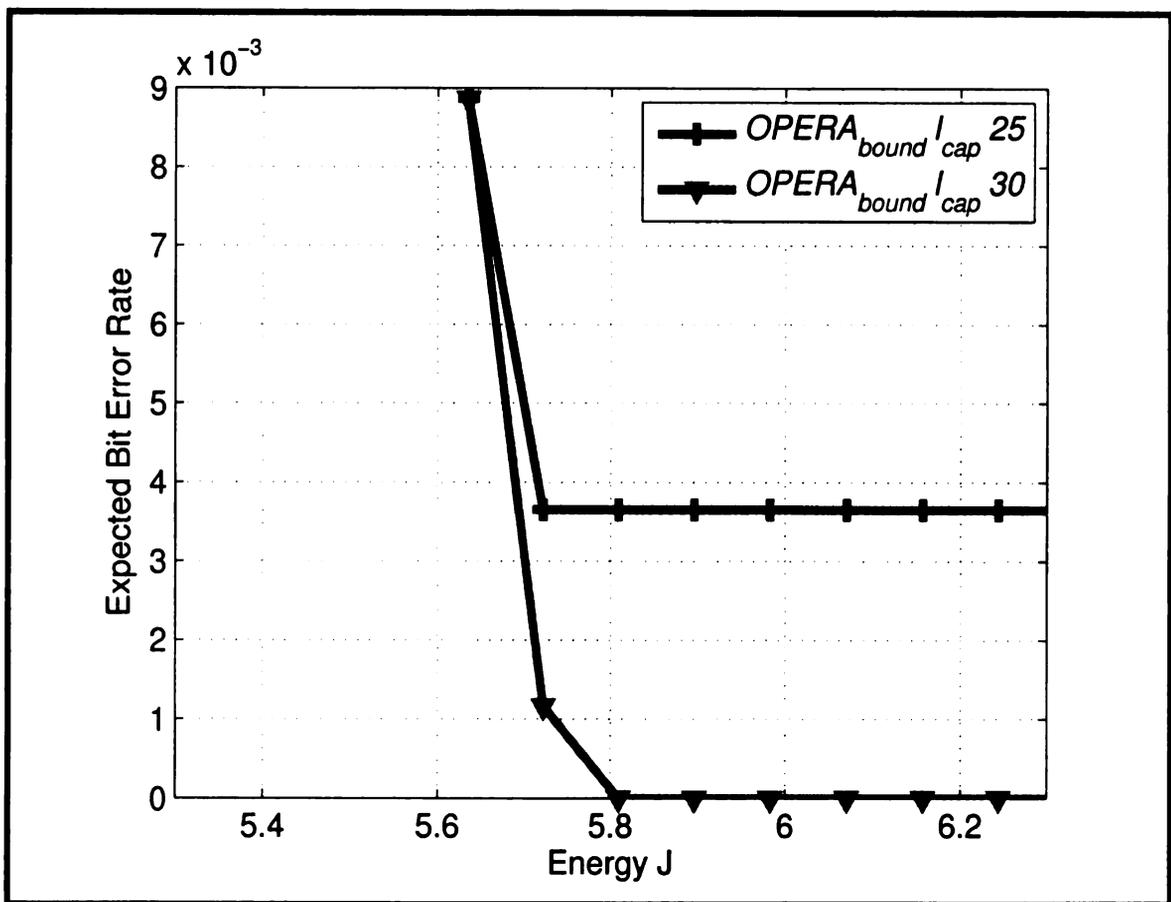


Figure 4.9. $OPERA_{bound}$ with $l_{cap} = 30$ and $l_{cap} = 25$, $R_N = 0.50$

Figure 4.10 shows the performance of $OPERA_{rc}$ as compared to OPERA. We see that $OPERA_{rc}$ consumes lesser energy than OPERA initially. This energy saving is due to lesser number of transmissions by the nodes closer to destination, though, price paid is the loss in decoding performance at the destination node. As in-network processing is increased, $OPERA_{rc}$ approaches OPERA in terms of throughput achieved for given energy budget with more fair energy distribution. The results also indicate the tradeoff in using higher coding rates, greater number of network partitions and throughput, with less energy usage by resource constrained nodes at the cost of some decrease in performance. Therefore, we can strike a compromise between desired re-

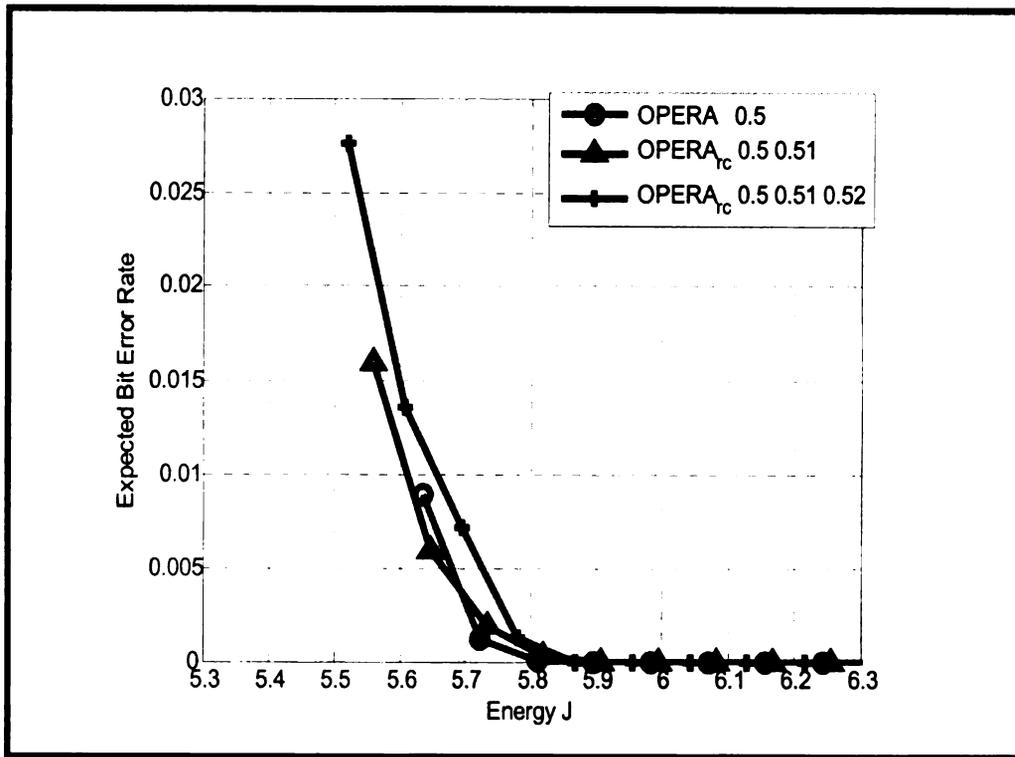


Figure 4.10. *OPERA* vs *OPERA_{rc}* for a) $R_{N_2} = 0.51$ and b) $R_{N_2} = 0.51, R_{N_3} = 0.52$

liability level and network energy resources with variation in network partitions and coding rates.

Bounded Rate Compatible *OPERA*

Figure 4.11 gives the performance curves for *OPERA_{brc}* with variation in l_{cap} . The plots show that it is practicable to trade-off distortion/reliability with network lifetime by variation in both the puncturing rates as well as maximum processing individual sensor nodes can perform. When we have lower l_{cap} and higher coding rates, packet errors may persist despite increase in intermediate node decoding.

In such a scenario, packets eventually received at the collector may have very little information utility. On the other hand, when l_{cap} is increased and lower coding rates

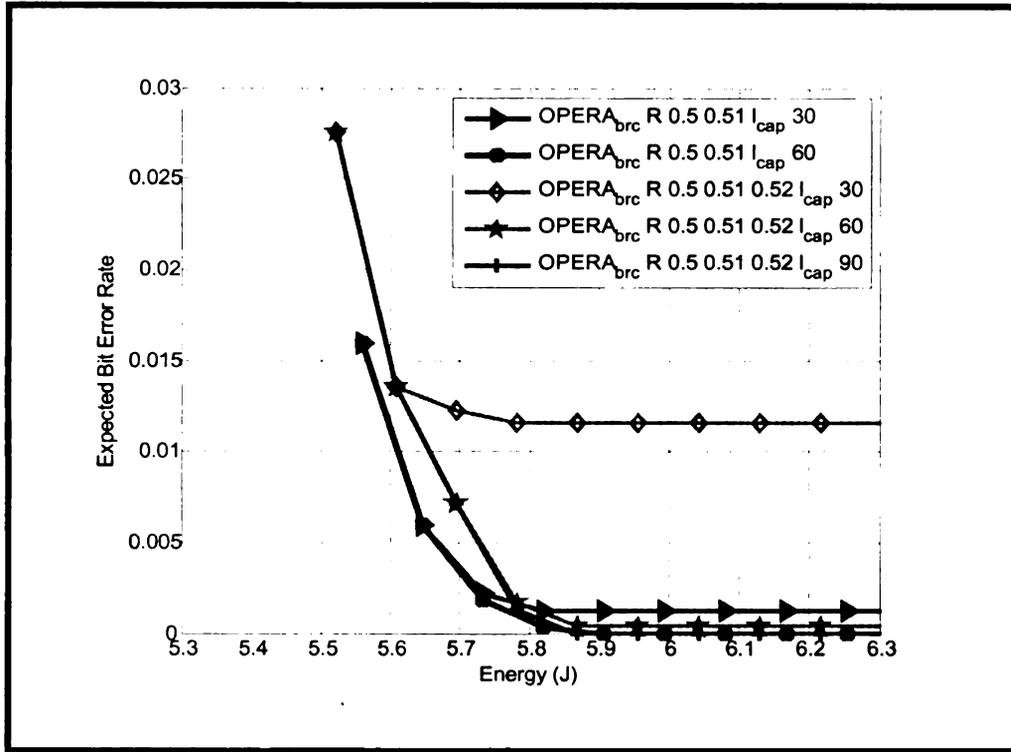


Figure 4.11. $OPERA_{brc}$ with $R_{N_2} = 0.51$, $l_{cap} = 30$, $l_{cap} = 60$ and $R_{N_2} = 0.51$, $R_{N_3} = 0.52$, $l_{cap} = 60$, $l_{cap} = 90$

are used in network partitions, data reliability can be increased significantly at the expense of greater energy usage. Thus, in such scenarios, even though throughput for sensor network is improved, price paid is a reduced network lifetime. The curves given here provide us the flexibility to select intermediate operating points. For a real sensor network deployment, event based operating point selection can be employed. When a significant event is sensed, lower coding rates at intermediate nodes with higher l_{cap} can be used. When the network is inactive, we can increase coding rates and/or decrease l_{cap} for enhanced network lifetime.



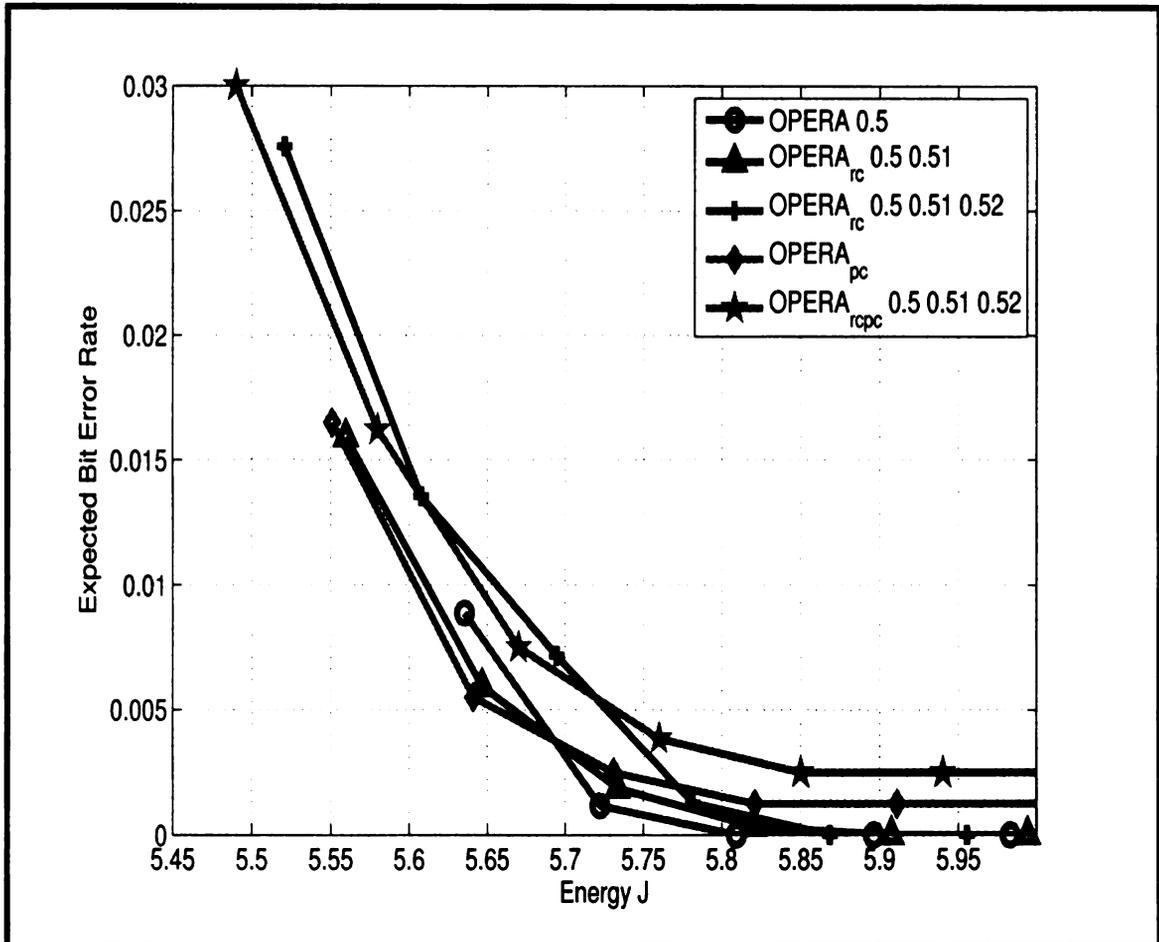


Figure 4.12. OPERA vs 1) $OPERA_{rc}$ for a) $R_{N_2} = 0.51$ and b) $R_{N_2} = 0.51, R_{N_3} = 0.52$, 2) $OPERA_{pc}$ & $OPERA_{rcpc}$ for a) $P_{N_1} = 1mW, P_{N_2} = 0.98mW, P_{N_3} = 0.96mW$ b) $P_{N_1} = 1mW, P_{N_2} = 0.98mW, P_{N_3} = 0.96mW, R_{N_2} = 0.51, R_{N_3} = 0.52$

4.5.5 OPERA with Power Control

We use identical partitions for $OPERA_{pc}$ as in $OPERA_{rc}$ with $R_N = 0.50, P_{N_1} = 1mW, P_{N_2} = 0.98mW$ and $P_{N_3} = 0.96mW$. Figure 4.12 compares the performance of OPERA with $OPERA_{pc}$. We see that though there are some energy savings using power control, the price paid is degradation in decoding performance. We further provide results in the figure when power control is used in conjunction with rate compatible puncturing of LDPC codes: $OPERA_{rcpc}$, taking $P_{N_1} = 1mW, P_{N_2} =$

$0.98mW, P_{N_3} = 0.96mW, R_{N_1} = 0.50, R_{N_2} = 0.51, R_{N_3} = 0.52$. Again, we see some power savings though the reliability of information content also suffers. Thus, based on the application demands & network conditions, we can switch between $OPERA_{rc}, OPERA_{pc}$ & $OPERA_{rcpc}$ to achieve fairness.

CHAPTER 5

Distributed Processing Allocation

In previous chapters, we presented a scheme that provides progressive reliability to WSN data as it traverses the multi-hop network. Though optimal, it assumes complete topological knowledge of the entire network. Such knowledge may not be available in real-world sensor network deployments. Thus, we are interested in placement of decoding iterations, in a distributed fashion, at the individual relays that achieves the highest decode and forward rate. Particularly, we are looking for a distributed progressive error recovery scheme that satisfies the following goals:

- Maximizes the throughput at the destination node in a distributed fashion
- Each node performs at least a single iteration
- Provides a decent approximation to the centralized counterpart: OPERA
- Minimal transmission and computational overhead on the individual sensor nodes
- Makes minimal or no assumptions regarding knowledge of the end-to-end path
- Offers a fair amount of asynchrony in distributing the processing budget
- Applicable to any iteratively decodable channel code

5.1 Pair-wise D-OPERA

For the iteration assignment problem, we are looking for an algorithm that does not require complete network knowledge and works fairly asynchronously to provide maximum reliability at the destination node. A simplistic approach can be to distribute

iteration budget equally among all the nodes. That does not necessarily maximize throughput, especially, when the variance among individual channel error rates is high. On the other extreme, giving individual nodes control of their iteration assignment values over a domain may result in each node choosing the maximum value for enhanced throughput which renders the finite iteration assignment problem irrelevant. Therefore, we are looking for a scheme that both ensures fair amount of asynchrony with increased throughput at the destination along with optimal use of nodes' computational resources.

We propose a scheme in which nodes compute iteration assignments in a pair-wise fashion using a Leader-Follower relationship. We assume that each node maintains an estimate of the error conditions of its associated channel based on some network parameter such as Link Quality Indicator, which is normally the case for WSNs [16]. The proposed D-OPERA scheme uses a three step approach. In the initial phase, nodes pair in Leader-Follower relationship. Leader nodes then solve a local minimization problem and obtain estimates for iteration assignment. The resulting follower node iteration assignment is then transmitted to the follower node in the last step. Figure 5.1 shows this diagrammatically.

In order to cater for dynamism in the network, each follower node periodically reports its channel error ϵ_S probability to the maser node. Leader node compares both $\Delta\epsilon_S$ and $\Delta\epsilon_M$ against a preset threshold ϵ_{TH} (where $\Delta\epsilon = |\epsilon_{new} - \epsilon_{old}|$). If either $\Delta\epsilon_S$ or $\Delta\epsilon_M$ or both exceed ϵ_{TH} , iteration assignment is carried out again. The periodic report frequency for the follower node can be designed based on the conditions in which nodes would be deployed and desired reliability levels.

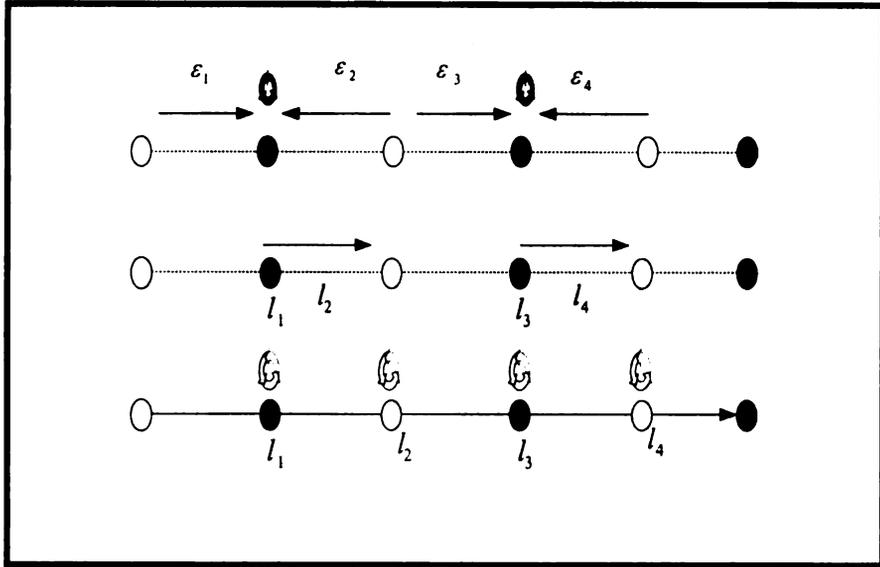


Figure 5.1. D-OPERA Pair-wise iteration assignment

5.1.1 Pairing

For a given flow, we group the nodes in pairs. Starting from the base station, the first node encountered is labeled as follower, second leader, and so on till all or all except a single node is left. Each pair is given a local budget of $\frac{2\Gamma(\bar{l})budget}{N-2}$ iterations with any leftover node getting $\frac{\Gamma(\bar{l})budget}{N-2}$ iterations. The pairing of nodes does not require complete network knowledge and can be achieved with quite an ease. The only information required in the process is number of nodes over the end-to-end path and total end-to-end iteration budget. We assume that information about total nodes is available through an underlying routing scheme which is usually the case for multi-hop wireless sensor networks [16]. Total iteration budget can either be passed to nodes during individual link establishment phase, or programmed in individual nodes prior to network deployment.

A side issue in distributed iteration assignment can be when iteration budget is not

completely divisible by $N - 2$ and leaves a remainder. This is handled in our work by rounding to nearest lower integer and adding any leftover iterations to the leader node closer to the data originator node, assuming data having greater survivability chance if provided further reliability closer to the originator than destinations, where errors may have accumulated to an extent that may lead decoding ineffective.

5.2 Local Minimization Problem

For D-OPERA, each leader node now solves a local minimization problem in which it attempts to minimize the local bit error rate with the condition that iteration assignment is within the local budget constraint. Mathematically, it solves:

Minimize

$$D_{loc} = P_e(\epsilon_L, l_L) * P_e(\epsilon_F, l_F)$$

Such That

$$l_L + l_F = \frac{2\Gamma(\bar{l})}{N-2}$$

Or

Minimize

$$D_{loc} = f(\epsilon_L, l_L) * f(\epsilon_F, l_F)$$

Such That

$$(l_L + l_F) = \Gamma(\bar{l})_{local}$$

We assume that the function $f(\epsilon, l) : \mathfrak{R} \rightarrow \mathfrak{R}$ is convex and twice differentiable [21].

We interpret this as the problem of allocating a single resource, with a fixed total amount $\Gamma(\bar{l})_{local}$ to two, otherwise, independent activities.

The problem can be solved using elimination. Thus, we have:

Min-

imize

$$f_L(\Gamma(\bar{l})_{local} - l_F, \epsilon_L) * f_F(l_F, \epsilon_F)$$

Subject to

$$\mathbf{1}^T \bar{l} = \Gamma(\bar{l})_{local}$$

$$l_L > 0$$

$$l_F > 0$$

where $\mathbf{1} \in \mathfrak{R}^{1 \times 2}$ is a unit vector with $rank \mathbf{1} = 1$ guaranteeing at least a single solution. Though standard elimination techniques (see for example [7]) can be used for the solution of minimization problem, they maybe computationally cumbersome for energy starved sensor nodes in terms of function evaluations and convergence properties. We instead turn to Newton's method for constrained optimization that provides faster convergence with quadratic approximation of the function D_{loc} that has advantages in terms of computational complexity with high accuracy and very fast convergence when the algorithm is initialized with good initial values that are close to optimal. Details of Newton methods can be found in any standard text (for example [7]) and are not mentioned due to brevity concerns .

5.2.1 Warm Start and Algorithm Convergence

The resource constrained nature of sensor nodes asks for low algorithm computational overhead and fast convergence. Faster convergence can be achieved in case of Newton's method when initialized to suitable values of the optimization variable. We

consider four approaches for providing D-OPERA a warm start:

D-OPERA Unitary start: Both the nodes start with a single iteration

D-OPERA Equal start: Both the nodes start with equal number of iterations

D-OPERA Max-Min start: One of the nodes starts with only one iteration, whereas, the other starts with $\Gamma(\bar{l}_{local}) - 1$, based on the associated channel error rate with $\epsilon_L \geq \epsilon_F$ implying $l_L \geq l_F$ and vice-versa.

D-OPERA Proportional start: proportional assignment based on the error rates with $l_L = \frac{\epsilon_L}{\epsilon_L + \epsilon_F} \Gamma(\bar{l}_{local})$ (rounded to lower integer value). For the case with a Leader/Follower node getting zero iteration, a single iteration is borrowed from the Follower/Leader node to ensure non-zero assignment.

For all these approaches, we note down the resulting error performance, number of function valuations and the algorithmic runs required to reach to a convergent solution. The resulting performance curves are presented in subsequent sections.

5.3 Results

For the simulation setup, we keep the assumptions regarding code choice and sensor notes as for the centralized case in chapter 5. We present both the cases here, first when we are given only a pair of nodes to be assigned decoding iterations and secondly, an extension of the proposed technique over multiple flows.

5.3.1 D-OPERA for a Pair

We consider 100 realizations of a Leader-Follower node pair in cascade over the end-to-end line network (Figure(4.1)) with each realization resulting in different channel error probabilities for the pair. The results are averaged over all the realizations. Leader node executes the algorithm and computes the iteration assignment for the pair.

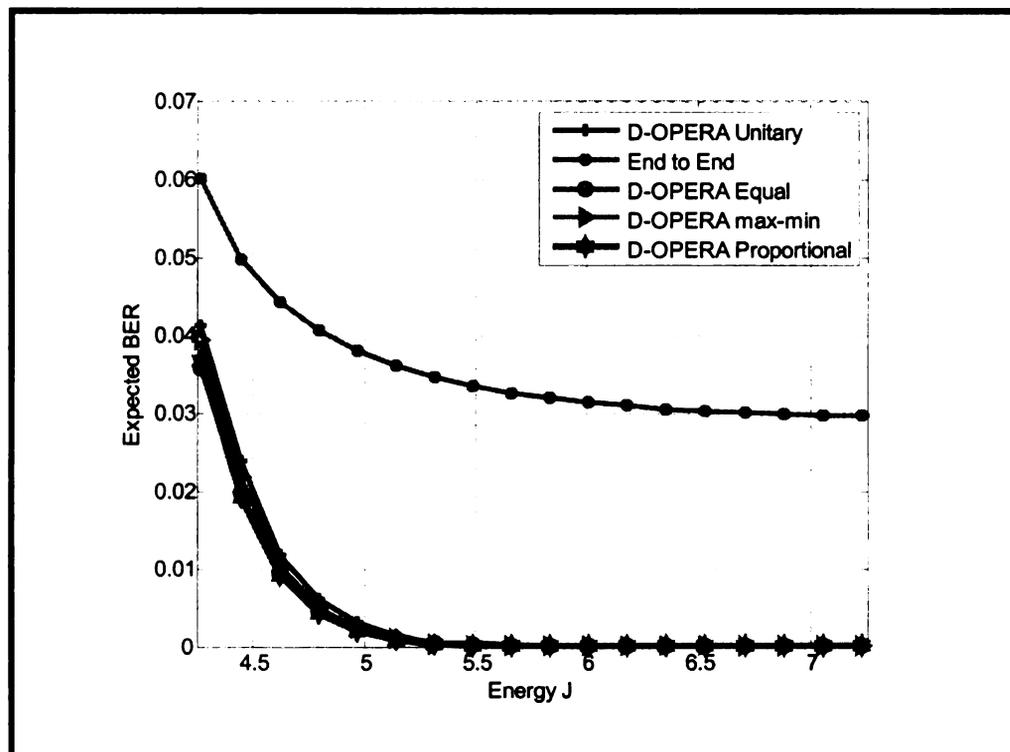


Figure 5.2. Expected Bit Error Rate vs Energy for D-OPERA

Figure 5.2 shows the performance of our scheme for both D-OPERA with all the initialization options and end-to-end channel coding. The results clearly indicate that D-OPERA outperforms end-to-end channel coding by a considerable margin with best performance achieved with D-OPERA Proportional.

Figure 5.3 and Figure 5.4 analyze the convergence characteristics of the proposed

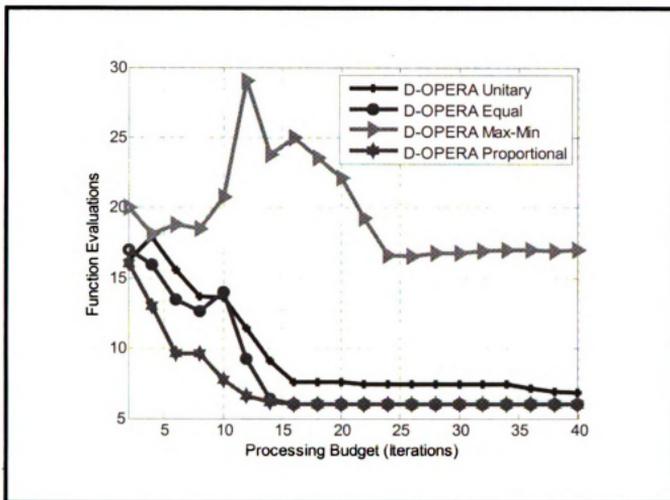


Figure 5.3. Processing Budget vs Functional Evaluations of D for D-OPERA

solution with differing warm starts. We see that both in terms of algorithmic iterates, as well as function evaluations, best convergence characteristics are achieved when the algorithm is initialized with proportionate number of iterations for both the nodes.

5.3.2 Multiple flows D-OPERA

We consider a wireless sensor network with 150 nodes spread over $10m \times 10m$ rectangular grid according to a random distribution. We place the base station at coordinates $(5, 5)$. The transmission range of each individual sensor node is limited to a maximum transmission range of $r = 2m$. Since most widely used sensor network routing protocols are some form of shortest path routing algorithm [16], we assume that at any given time, the routes from individual nodes to the base station form a tree rooted at

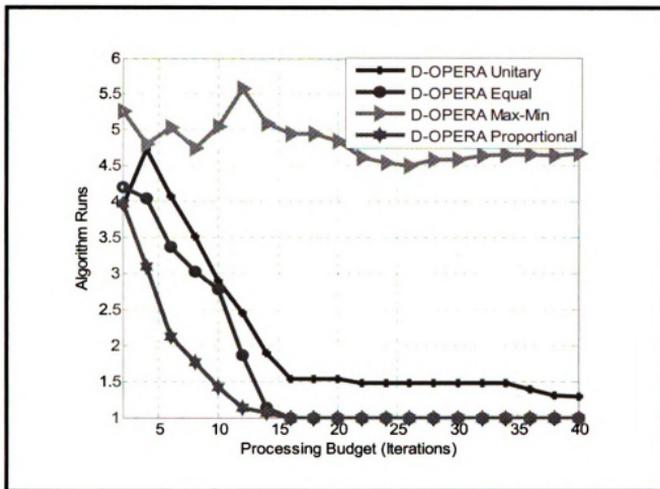


Figure 5.4. Processing Budget vs Algorithmic Runs

base station. $\Gamma(\bar{l})_{budget}$ is varied from 6 to 40 per flow. The results in Figure 5.5 are after complete decoding at the destination node for both OPERA and D-OPERA. The results clearly show that D-OPERA provides a close enough approximation to its centralized counterpart, OPERA. The curves also reaffirm our observation in previous chapters concerning significance of partial decoding at intermediate hops. When no decoding is performed at intermediate nodes, accumulation of errors grows exponentially. In such situation, most of the packets received at the destination may have very little or no information utility. On other extreme, when sufficient decoding is employed at all intermediate nodes, data reliability increases significantly at the cost of energy consumption. Our technique allows to fine-granularly operate at number of intermediate points with variation in $\Gamma(\bar{l})_{budget}$.

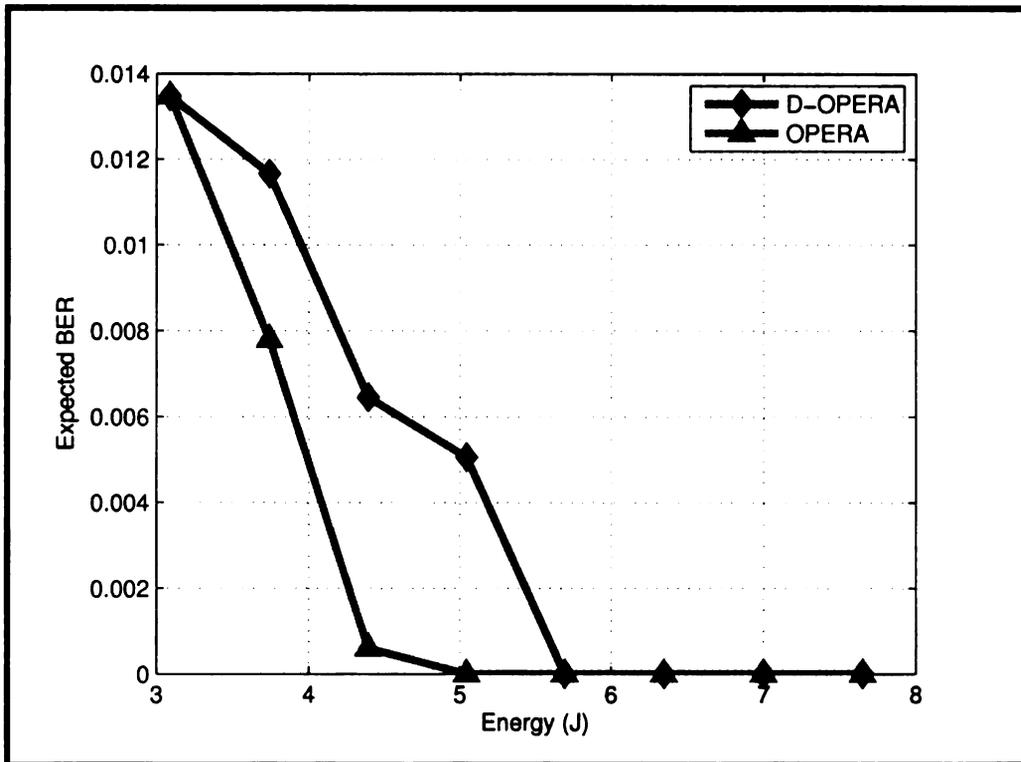


Figure 5.5. D-OPERA vs OPERA for Multiple Flows

CHAPTER 6

Progressive Error Recovery with Path Diversity

6.1 Introduction and Motivation

We present scenarios in which the proposed partial decoding scheme can provide significant performance benefits as compared to the end-to-end counterpart when multiple paths are available leading to the destination node. We develop our framework in conjunction with directed diffusion paradigm [17], though the scheme is equally applicable to any other routing scheme in place with minor modifications.

We investigate the setting when there is asymmetry in the significance of data generated by individual nodes due to differing reliability demands set forth by each type of data. An example of such an application can be in security & surveillance related scenarios where events such as presence of fast moving objects in the vicinity of sensor maybe of more interest than slow ones, or vice-versa. In such a case, we intend to provide one of the events greater reliability than the other along-with minimal energy consumption. Our scheme caters for that by introducing selective budgeting of network resources for each type of event. The proposed architecture is especially viable for Wireless Multimedia Sensor Networks (WMSNs) [3] where, there can be multiple streams of data originating at a source node, and each may require different level of reliability. For instance, for environmental monitoring applications, we may have both acoustic and video data feeds that need to be transmitted back to the base station. Likewise, in some surveillance applications, network designer maybe willing to dedicate more network energy resources to audio data than video or vice-versa. Similarly, applications where there can be variation in sensor data reliability demands such as monitoring and sensing, storage of potentially relevant

activities, traffic avoidance and control, advanced health care delivery etc [3] are good contenders for our proposed framework.

We further show that the proposed framework provides flexibility to operate at much higher error rates with greater energy efficiency. MDR provides an added advantage in that, unlike traditional reliability schemes where only one path is available to reach back to the destination node, providing multiple paths for sensor data offers greater network lifetime by introducing a delay in network partitioning [34].

6.2 Background

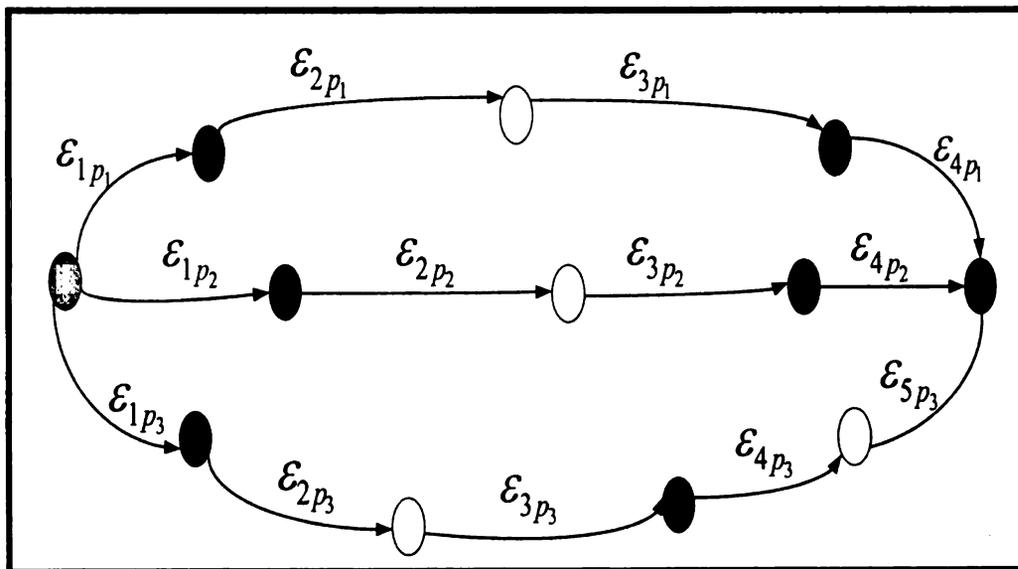


Figure 6.1. A multi-hop wireless sensor network with multiple paths to destination node

6.2.1 Directed Diffusion of Interest

For the scope of our work, we assume that a routing scheme such as directed diffusion [17] is already in place. In directed diffusion, a destination node initiates a request for data by flooding interests for named data. It uses attribute-value pairs in order to name data [17]. Each interest contains attributes such as *type* of interest, timestamp, *active duration & rectangular region* over which event is expected to take place. The interests are passed over the network in a manner that each node knows only its immediate neighbor. Thus, interest may reach the source node through multiple paths. Hence, as indicated in Figure 6.1, several paths are available for sending back the data to the destination node which are selectively reinforced based on network application and demands. Our proposed methodology employs these paths to route the data back to destination node.

6.3 Multi-Path Distributed Reliability Framework

6.3.1 Phase I: Data Partitioning

We propose a framework that provides reliability to wireless sensor data maintaining energy efficiency during the process. This is achieved in two phases: in first phase, data is partitioned into multiple streams based on its significance/genre. On the basis of partitions thus obtained, we decide on processing budget at each intermediate node, for the path assigned to that data stream. In the second phase, this budget is distributed using an optimization algorithm carried out at selected intermediate

nodes. Both the phases are presented in subsequent subsections.

In the first phase, nodes partition the data into multiple data sets based on its significance. The decision on significance can either be source based or destination based. For the destination based case, destination node initiates an interest for an event that is passed on over the network. The interest may reach the node through multiple paths. We propose a new field in directed diffusion interest: *'importance'* specifying significance of the data. The field can be adapted to have multiple values based on application in which sensor network is deployed and the precise details are left for network designer. The information passed by *'importance'* field is used by the source nodes to decide the local decoding budget for each pair of nodes over the end to end path.

In source based scheme, the decision on significance of data is local to the data originating node.

6.3.2 Phase II: Multi-Path Distributed Iteration Assignment

Once the source node is decided on data genre and significance level it belongs to, it assigns the resulting data stream to an end-to-end path. We assume that each path has equal likelihood to be chosen for a type of data. After path selection and hand shaking procedures, intermediate nodes decode the data with finite number of decoding iterations as it traverses the multi-hop network. A question though remains on the exact number of decoding iterations carried out by each node. We address this by nodes computing iteration assignments in a pair-wise fashion having a Leader-

Follower relationship.

In our scheme, each node maintains an estimate of the error conditions of its associated channel based on some network parameters such as Link Quality Indicator [16]. In the initial phase, nodes pair as Leader-Follower. Pairing of nodes does not require global knowledge and can be achieved with quite an ease during link establishment phase. Starting from the source node, first node encountered is labeled as Lead, second Follower and so on till all or all except a single node is left.

Assuming Γ a summation operator, each pair has a local pair budget of $\Gamma(\bar{l})_{pairp_i}$ determined based on significance level of data stream that needs to be distributed within Lead & Follower nodes such that resulting distortion D_{pairp_i} is minimum for that set of nodes. Each Lead node asks Follower node for their channel error probability ϵ_F . Lead node uses both ϵ_F and ϵ_L to formulate a local minimization problem:

$$\text{Minimize} \quad D_{pairp_i} = f(\epsilon_L, l_L) * f(\epsilon_F, l_F)$$

Such That

$$l_L + l_F = \Gamma(\bar{l})_{pairp_i}$$

$$l_L > 0$$

$$l_F > 0$$

where

$$\epsilon_1 * \epsilon_2 = \epsilon_1(1 - \epsilon_2) + \epsilon_2(1 - \epsilon_1)$$

We assume that the function $f(\epsilon, l) : \Re \rightarrow \Re$ is convex and twice differentiable [23]. We use Newton's method for constrained optimization to solve the minimization problem at the Leader node due to its advantages in terms of computa-

tional complexity and faster convergence when the algorithm is initialized with good initial values [7]. For the given minimization problem, we initialize the algorithm with iteration values proportional to the channel error rates such that for the lead node, $l_L = \frac{\epsilon_L}{\epsilon_L + \epsilon_F} \Gamma(\bar{l})_{pair}$ (rounded to lower integer value) ensuring fast convergence within few algorithmic runs. Any leftover node without a pair in the pairing process is assigned $\frac{\Gamma(\bar{l})_{pair}}{2}$ iterations. We have already shown the pair-wise iteration assignment in Figure 5.1 for a single end-to-end path starting from channel error probability exchanges between Leader and Follower nodes, iteration assignments based on Newton's optimization and eventually, decoding at intermediate hops.

In order to cater for dynamic network conditions over multiple paths, each Follower node periodically reports its channel error probability ϵ_F to its Leader. The Leader compares both $\delta\epsilon_L$ and $\delta\epsilon_F$ against a preset threshold ϵ_{TH} (where $\delta\epsilon = |\epsilon_{new} - \epsilon_{old}|$). If either of $\delta\epsilon_L$ or $\delta\epsilon_F$ exceeds ϵ_{TH} , iteration assignment is recomputed at the Leader node and passed on to Follower. The rate at which Follower reports ϵ_F to Leader can be adapted based on conditions in which network is to be deployed and the reliability desired.

6.4 Simulation Setup

We consider a source node with multiple paths leading to base station setup through a multipath reinforcement scheme such as directed diffusion [17]. We assume three data streams originating at a source node each requiring different reliability level based on either destination based or source based partitioning. Source node specifies

the desired reliability level for each stream based on which level it belongs to. We assume that ϵ_{p_1} is the averaged equivalent error probability [22] over all the hops for end-to-end path p_1, ϵ_{p_2} for path p_2 and so on. Thus ϵ_{av} is the composite error probability for the network averaged over all the paths. We use multiple realizations of the end-to-end multi-hop links with variation in error probability such that number of nodes for each path is $N_{p_1} = 4, N_{p_2} = 5, N_{p_3} = 4$. Each realization results in a different individual link error probability such that there is minimal deviation for each path from average equivalent error probability. We assume all the nodes having uniform energy levels initially.

We set the information packet size to be $k = 1024$ bits which is encoded with a rate $R = 0.569$ code. We use an LDPC code with degree distribution polynomial $\lambda(\theta) = 0.207\theta^6 + 0.271\theta^2 + 0.522\theta^1$ [27], though, by concentration property of LDPC codes [27], the deductions made here are equally applicable to any randomly picked code from LDPC ensemble.

The transmission range of each individual sensor node is limited to a maximum transmission range $r = 2m$. We set $\epsilon_{min} = 0.005$ and $\epsilon_{max} = 0.08$. We take per bit transmit power $P_T = 1mW$, per second, for each node. We assume that if $\epsilon < \epsilon_{min}$, only one iteration is enough to decode the information bits which is usually the case [22] [21]. We further assume each node equipped with an Atmel Atmega128L processor and 2000:1 ratio between per bit transmission energy and computation energy spent per instruction [12]. The average number of ones per column in a parity check matrix of a (n, λ, ρ) code from ensemble of LDPC codes is taken as: $(\int_0^1 \lambda(\theta) d\theta)^{-1}$ [32]. Fossorier et al. [9] give a comparison of mathematical operations

required for one LDPC decoding iteration for various LDPC decoding algorithms including sum product algorithm [27]. This information is used to determine the computational energy spent by sensor nodes for a single LDPC decoding iteration. We use log-domain sum product algorithm for LDPC decoding due to its advantages in terms of computational complexity. All results are averaged over 100 runs. For the end-to-end case, the total energy is the transmission cost of delivering information bits reliably across the multi-hop network, whereas, energy costs for MDR involve both transmission and computations within the network.

At the destination node, decoding is performed for each data stream. The resulting bit error rate and energy levels are sampled such that there is one to one correspondence between energy levels for all the paths and the bit error rates thus achieved for each path during that run. The results are averaged to obtain expected bit error rate for corresponding energy level.

6.5 Results and Discussion

Figure 6.2 gives a performance comparison between proposed MDR scheme and end-to-end channel coding with variation in end-to-end channel error probability ϵ_{av} . The results clearly establish that the proposed MDR scheme outperforms end-to-end channel coding by considerable margin for level of reliability achieved for a given energy budget. The plots further indicate that for end-to-end approach, as the energy budget is increased, the corresponding decrease in bit error rate is relatively irregular till enough redundancy is not added to sustain the information bits over entire end-

to-end path, at which time though there is a sharper decrease, energy cost is also significantly higher. For MDR approach, rather than adapting the whole packet size for different reliability levels at much higher energy costs, we achieve the flexibility to operate at fixed packet sizes with greater energy efficiency by relatively in-expensive processing within the network, thus ensuring network lifetime maximization.

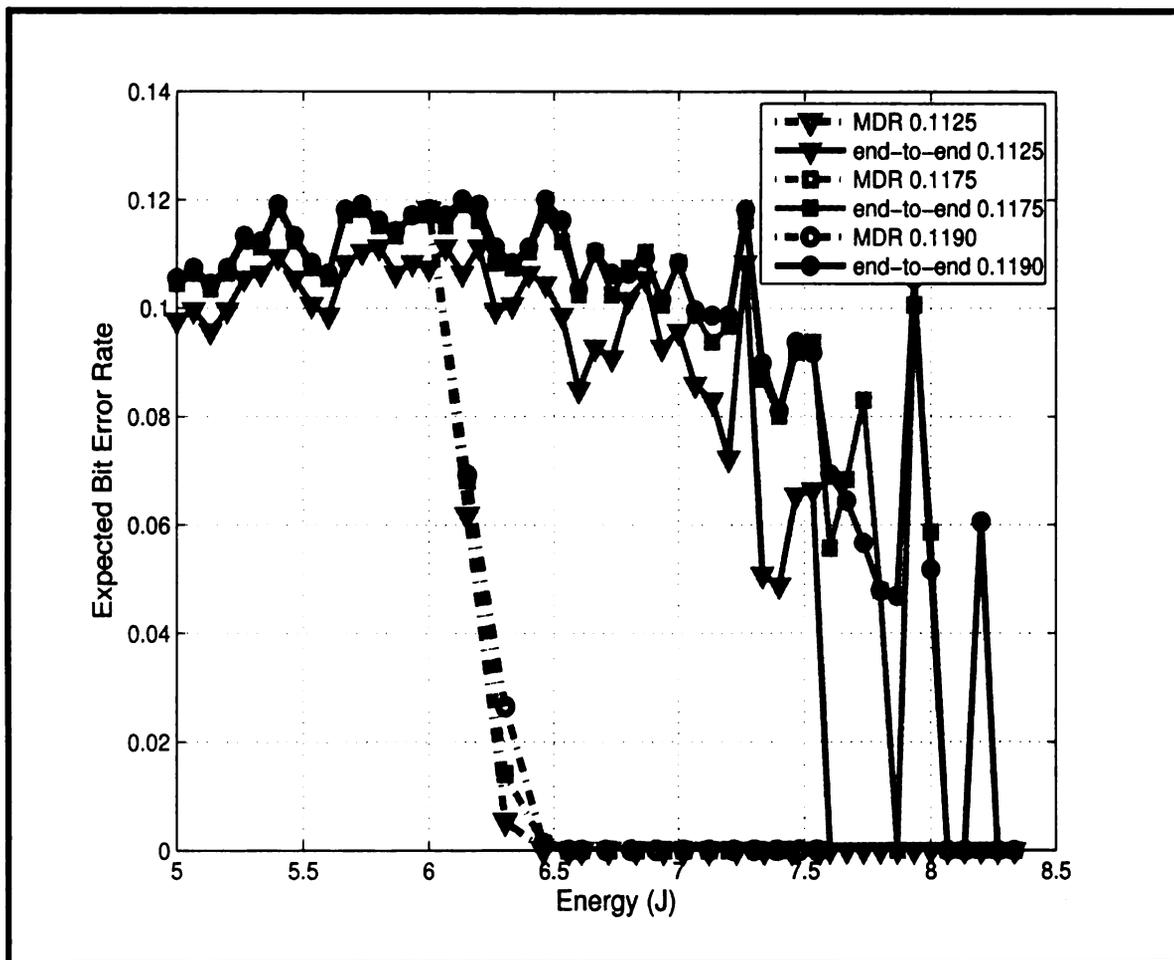


Figure 6.2. MDR vs end-to-end for $\Gamma(\bar{l})_{pairp_1} = 4$, $\Gamma(\bar{l})_{pairp_2} = 6$ and $\Gamma(\bar{l})_{pairp_3} = 8$, $\epsilon_{av} = 0.1125, 0.1175, 0.1190$

Figure 6.3 shows the variation of expected bit error rate for MDR scheme when pair-wise budget $\Gamma(\bar{l})_{pair}$ is varied for each data stream. We see that careful choice of $\Gamma(\bar{l})_{pair}$ is important in ensuring desired level of reliability. We intend to maximize

throughput at the destination node, but at the same time, do not want to create a partition within the network by draining out all the nodes on that path. The results further indicate the efficacy of MDR in allowing us to operate at end-to-end channel error rates as high as $\epsilon_{p2} = 0.1364$ and still maintaining energy efficiency.

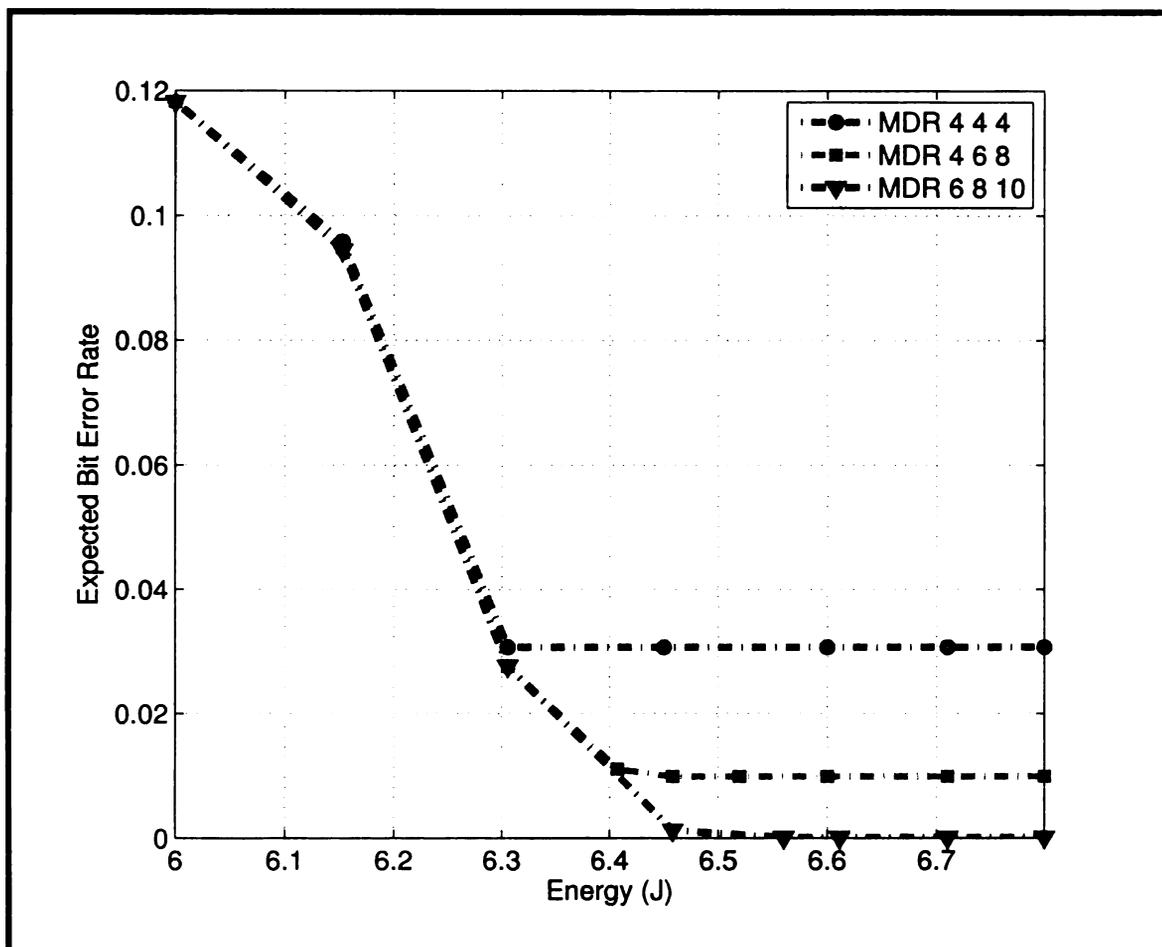


Figure 6.3. MDR vs end-to-end for i) $\Gamma(\bar{l})_{pairp_1} = 4, \Gamma(\bar{l})_{pairp_2} = 4, \Gamma(\bar{l})_{pairp_3} = 4$
ii) $\Gamma(\bar{l})_{pairp_1} = 4, \Gamma(\bar{l})_{pairp_2} = 6, \Gamma(\bar{l})_{pairp_3} = 8$ iii) $\Gamma(\bar{l})_{pairp_1} = 6, \Gamma(\bar{l})_{pairp_2} = 8,$
 $\Gamma(\bar{l})_{pairp_3} = 10, \epsilon_{p1} = 0.1190, \epsilon_{p2} = 0.1364, \epsilon_{p3} = 0.1190$

The plots in Figure 6.2 & Figure 6.3 further show that it is feasible to tradeoff complexity/energy usage with distortion/reliability for a given sensor network. The two extremes of the energy/distortion curves in Figure 6.3 indicate the performance

that is achieved when two extremes of in-network processing is employed. When no decoding is performed at intermediate nodes, the accumulation of errors is exponential. Thus, any packets received at the destination node may have very little or no information utility. On the other hand, when complete decoding budget is distributed among all the pairs over all the paths, the information reliability is maximum, though, network energy consumed also increases proportionally resulting in decreased network lifetime. Our proposed scheme allows to fine-granularly operate at intermediate points by adjusting the functional demands of the network such that a balance is maintained between desired reliability level and network lifetime. Again, this can be achieved by adapting the pair budget assigned to individual streams over the network lifetime.

CHAPTER 7

Application: Still Images

7.1 Preliminaries and Motivation

In this chapter, we investigate the performance of proposed progressive error recovery paradigm in conjunction with still images.

As of today, most of the wireless sensor networks deployed handle data that is of scalar physical phenomena such as pressure, humidity, temperature, location etc [3]. Wide scale availability of CMOS devices such as CMOS cameras and audio sensors has made it possible to integrate large selection of image and audio processing applications with sensor networks. Though these developments have opened wide range of possibilities, they have also brought forth various research challenges including provision of data reliability with minimal overheads.

In this work, we utilize the progressive error resilience framework in order to provide error resilience to sensor visual observations as they traverse the multi-hop network towards the destination node. We propose to use partial channel decoding of visual data as it approaches the destination node. Unlike some of the prior existing work where full channel decoding/encoding was conducted under a Network Embedded FEC (NEF) framework [41], here we use progressive error recovery paradigm discussed in previous chapters. We show that the proposed paradigm, when used in conjunction with visual data, significantly enhances received image quality as compared to prevalent reliability schemes for a given network energy budget. We further show that little bit of processing at intermediate nodes can greatly enhance the peak signal to noise ratio (PSNR) quality of wireless visual data without compromising on energy efficiency. We illustrate that our approach is suitable for low-power visual

sensor networks through rigorous simulations.

7.2 Simulation Setup

We conduct simulations based on both OPERA and D-OPERA presented in chapters 4 and 5. In order to support image transfer, additional features have been added to the application that enables the fragmentation and reassembly of the compressed image files. We use standard image sequences ‘Boats’ and ‘Lenna’ for our simulations. The images are JPEG2000 encoded, packetized, LDPC channel encoded and transmitted through a cascade of binary symmetric channels. A reverse process including LDPC decoding, re-framing and JPEG2000 decoding is conducted at the receiver. Due to the nature of our problem, for the scope of this work, we do not take source-encoding complexity into consideration and assume that sensor nodes are capable of successfully encoding the images. We use Peak Signal to Noise Ratio (PSNR) in order to measure quality of received images.

The captured image at the source node is first encoded using a JPEG2000 encoder with a compression ratio of 20:1. The JPEG2000 encoded bit stream is then packetized with $k = 1024$ information bits in each packet and fed to an LDPC encoder with a degree distribution polynomial: $\lambda(\theta) = 0.207\theta^6 + 0.271\theta^2 + 0.522\theta^1$ and code rate R . The resultant packets are then transmitted over multi-hop network with associated channel error probabilities ϵ .

In order to estimate the total energy consumption E_{total} of the network for transporting a data packet over end-to-end path, we set transmission power of each node

$P_T = 1mW$ with 1:2000 ratio between transmission cost and computation cost [12]. Fossorier et al. [9] give a comparison of mathematical operations required for LDPC decoding, per iteration, for various LDPC decoding algorithms including sum product algorithm. We use log-domain sum-product algorithm due to its advantages in terms of computational complexity [27]. The average number of ones in an LDPC parity check matrix is taken as $(\int_0^1 \lambda(\theta) d\theta)^{-1}$ [32]. We assume that each node is sending message back to a central coordinator over a multi-hop WMSN with number of nodes $N=4$ over the end-to-end path. We assume each node equipped with Atmel Atmega128L processor. For the statistical model, we assume $\epsilon_{min} = 0.005$ and $\epsilon_{max} = 0.08$. We use different realization of the network for each run such that average end-to-end equivalent error probability ϵ_{eq} remains constant. All the results are averaged over hundred runs.

7.3 Results and Discussion

Figure 7.1 shows the performance of proposed DPERA scheme in comparison to end-to-end channel coding with $E_{total} = 6.5J$ for both end-to-end and D-OPERA, with $\Gamma(\bar{l})_{pair} = 4$. We see that D-OPERA performs significantly better as compared to the conventional end-to-end channel coding in terms of received image quality.

Figure 7.2 highlights this further. We see that for end-to-end equivalent channel probabilities as high as $\epsilon_{eq} = 0.14$, D-OPERA shows excellent PSNR characteristics. In comparison, for end-to-end scheme, in order to achieve similar levels of reliability, we may need much bigger packet lengths (and hence more redundant transmissions)

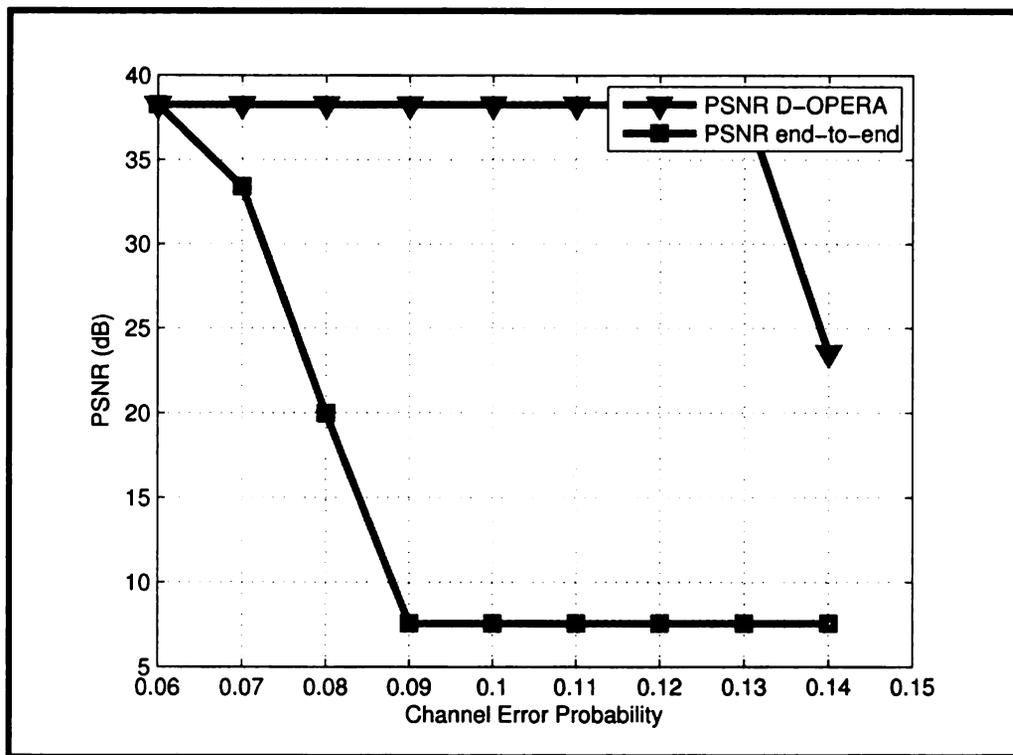


Figure 7.1. Average PSNR for D-OPERA with $\Gamma(\bar{l})_{pair} = 6$ and end-to-end for $R = 0.5$ LDPC encoded JPEG2000 image

leading to significantly higher consumption of energy. Even if more redundancy is added for end-to-end scheme, end-to-end error probabilities of the order of 10^{-1} maybe above the capacity of even best available channel codes.

For the proposed D-OPERA architecture, further savings in terms of energy can be achieved using prioritization of the data, such that data having greater significance is assigned a path with higher $\Gamma(\bar{l})_{pair}$ and lower significance data lower $(\Gamma(\bar{l})_{pair})$. For instance, in JPEG2000 progressive coding of image using multiple layers [37], some layers may require higher protection against channel errors than others based on their energy efficiency and contribution to final PSNR. In this work, we restrict our discussion to uniform $\Gamma(\bar{l})_{pair}$ over the end-to-end path.

Figure 7.2. JPEG2000 encoded ‘boats’ image with $\epsilon_{eq} = 0.14$ (clockwise from left) i- original image after source encoding ii- end-to-end channel coding iii- OPERA with $\Gamma(l)_{budget} = 6$ iv- D-OPERA with $\Gamma(l)_{pair} = 6$

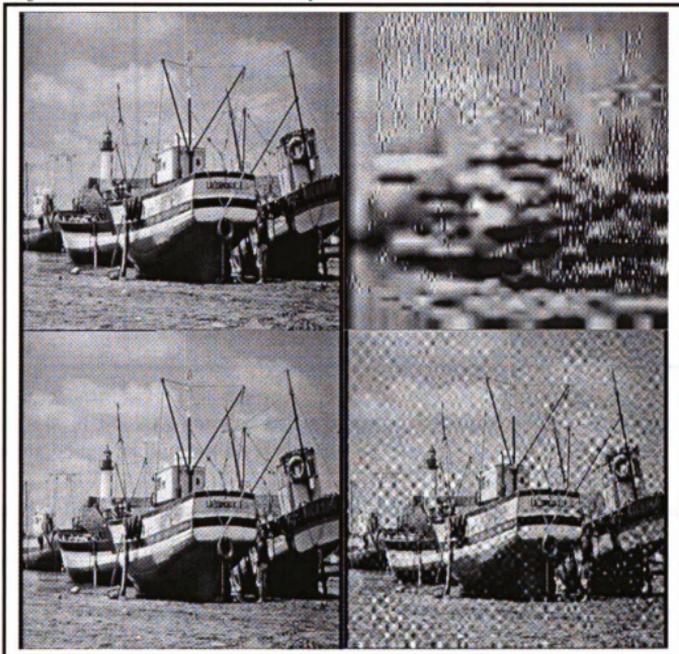


Figure 7.3 gives the performance curves for proposed OPERA and D-OPERA schemes for JPEG2000 encoded Lenna image sequence over a line network with $N = 4$ and an LDPC code rate of $R = 0.5$. The curves visibly show the performance gains achieved in using OPERA/D-OPERA. For an iteration budget of only six iterations to be distributed over a pair of nodes, the gain in peak signal to noise ratio is tremendous for progressive error recovery. For end-to-end scheme, we start getting decoding

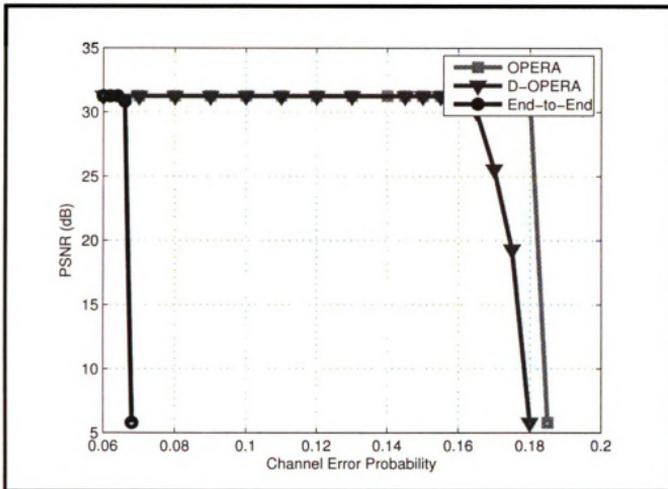


Figure 7.3. JPEG 2000 encoded Lenna image with PERA and D-OPERA for a pair decoding budget of 6 iterations $R = 0.5$

failure for JPEG2000 encoded Lenna at approximately 7% channel error probability.

In contrast, we do not encounter decoding failure for error rate as high as 14% and 18% for D-OPERA and OPERA, respectively, with the assumption that ϵ remains below code capacity [28] on individual hops.

For the progressive recovery paradigm, the choice of $\Gamma(\bar{l})_{pair}$ is an issue that requires vigilance for the network designer & should be chosen based on the conditions in which network is expected to operate. For instance, too high $\Gamma(\bar{l})_{pair}$ may lead to faster depletion of node resources, whereas, too low $\Gamma(\bar{l})_{pair}$ may not provide desired reliability. Hence, a balance should be struck while setting this parameter.



Figure 7.4. JPEG2000 encoded 'lenna' image with (clockwise from left) i- original image after source encoding ii- end-to-end channel coding with $\epsilon = 0.07$ iii- PERA with $\epsilon = 0.18$ iv-D-OPERA with $\epsilon = 0.143$

CHAPTER 8

Application: Wireless Video

8.1 Introduction

Wide scale availability of low-cost hardware such as CMOS cameras and microphones has fostered the development of sensor networks to carry multimedia content [3]. For instance, Crossbow's Imote2 multimedia board [1] adds multimedia capabilities to the Imote2 platform for Wireless Sensor Networks (WSNs) allowing for capturing images, video as well as audio for playback. This opens door for a variety of applications for WSNs along with posing certain challenges in terms of reliably routing the multimedia content from sensor nodes back to the base station. An example of such an application can be in surveillance and security related scenarios where video data captured by a remotely deployed camera mounted on a sensor node might need to be reliably sent back to the central base station. An inherent problem in applications involving video data is the amount of data generated along with ensuring minimal expenditure of energy in routing it back to the base station. Using a single path to the destination would fast deplete resources on that path, resulting in early network partitions and lesser than expected network lifetime.

In [20], Puri et al. discuss the challenges faced by any video transmission scheme for wireless sensor networks. They conclude that a broadband network of wireless video sensors is subjected to three principal constraints: limited processing capabilities, limited power/energy budget and information loss endemic to the harsh-loss prone wireless communication environment. A related issue is the high transmission costs for energy ravenous sensor nodes. According to some studies, there is a 1:2000 ratio between transmission cost and computational cost [12] in WSNs, thus making a

strong case in favor of in-network processing. Any data reliability framework should keep these limitations under consideration. These limitations and the information losses call for robust coding algorithms, protocols and architectures that build error robustness within the WSNs.

Motivated by both error prone nature of links as well as sensor energy limitations, we provide an architecture that can be used in order to reliably transport video content in WSNs. The proposed framework ensures near optimal use of network resources as well as network lifetime maximization by avoiding partitions in the network due to over usage of few paths. Particularly, we propose a strategy to add reliability in WSNs using a three prong approach that builds on: a) multi-stream coding of video b) multipath transport and c) distributed progressive error recovery through partial recovery of sensor data at intermediate nodes (discussed later). To the best of our knowledge, no work exists in this regard with these three components working together in WSNs. We propose a solution that is best suited to multi-stream coding of video as prevalent in most modern video coding architectures [20]. The proposed architecture is distributed in nature such that no complete knowledge is assumed regarding end-to-end network conditions. Using LDPC codes, we compare the performance of the proposed framework with end-to-end channel coding and show that it significantly outperforms it in terms of energy efficiency for level of reliability thus achieved.

8.2 Background

In this section, we provide a quick overview on few of the concepts used in the proposed framework.

8.2.1 Video Coding Techniques

In order to understand the WSN video reliability problem, we take a quick glance on the nature of coding techniques available for coding video content. Possible candidates for WSN can be adapted form of either of layered coding (LC), multiple description coding (MDC) or distributed video coding (DVC).

A typical layered coder includes one base layer and one or more successive enhancements layers that can be used together to achieve a desired level of video resolution at the destination. The base layer has higher priority than the enhancement layers as the loss of base layer makes the information received from the enhancement layers useless [25].

Multiple description coding fragments a single media stream into M independent sub streams ($M > 2$) referred to as descriptions. These descriptions are then sent over multiple channels and combined at the receiver depending on the desired video resolution [40].

Predictive video coding (PVC) as used in H.26x or MPEG-x employs two modes in order to encode video: Intra-coding (I) mode exploiting the spatial correlation in the frame that contains the current block by using a block transform such as the discrete cosine transform (DCT), Inter-coding or motion compensated predictive (P)

mode that exploits both spatial and temporal correlation in the video sequence [20]. Thus, in PVC, the encoded video frames can be partitioned into an I-stream and a P-stream.

A more recent work on low complexity encoding of sensor video data is characterized by distributed video coding (DVC) which incorporates concepts from source coding with side information, creating an intra-coded I-frame along with a side information counterpart Wyner-Ziv (WZ) frame [20]. The DVC framework, using its PRISM architecture, promises to provide robustness, light source-encoder architecture and flexibility in distributing the computational burden of motion estimation between transmitter and receiver. We can notice here as well that data can be partitioned into multiple streams consisting both intra-coded as well as side information frames.

In summary, video coding techniques currently in vogue create multiple streams of data that may differ in their significance. Any video sensor reliability framework should consider these facts in the design of the technique.

8.2.2 Multipath Transport (MPT)

MPT has been studied at length for both wired and wireless data [19]. Multipath routing focuses on finding maximally disjoint paths from source to destination node. Various routing algorithms used in sensor networks such as directed diffusion, dynamic source routing (DSR), ad-hoc on demand distance vector (AODV) return multiple paths to the destination node [19]. Routing diversity-where packets are purposely sent on different routes to insure against the failure of a single route-can increase error

robustness in WSNs [13]. Thus, sensor energy limitations and multi-stream nature of generated video content makes an appealing case in favor of MPT for avoiding early network partitions through load balancing and provision of error resilience.

8.3 Multi-Path Multi-Stream Distributed Reliability

We discussed various video coding techniques that can be potentially adapted for use with WSNs. One thing is common in all of them, data can be partitioned into various streams. These streams can either be self sufficient in that only one of them can be used to decode the information (such as in MDC) or hierarchical in the sense that we need one of them in order to decode the second one. We propose a multi-stream multipath distributed reliability (MMDR) framework that is generic in nature such that any technique used for encoding, till the time it has streams, we are good. The framework uses a three prong strategy:

1. Source encode the video data into multiple streams and channel code the resulting streams
2. Assign the video to multiple paths for ensuring load balancing and error resilience
3. Use partial decoding at intermediate nodes through distributed progressive error recovery framework (discussed subsequently) to recover from channel induced errors

Figure 8.1 gives the details of the proposed MMDR setup in WSNs. In the initial phase, the video data is source encoded such that we attain multiple streams. These streams are then channel coded before assigning to various paths. In the subsequent subsections, we present the details on the components of the proposed architecture.

8.3.1 Prioritized Data Partitioning

The source encoded video is partitioned into M streams based on the significance of data towards decoding of video. e.g. assuming that we are using H.264 [25] for coding video, the encoded video frames are split based on whether they are intra-coded (I-frames) or inter-coded (P-frames). These streams are then channel encoded before they are fed to a path assignment block.

8.3.2 Path Assignment

We assume that certain multipath routing scheme such as directed diffusion is already working underneath that returns multiple paths to the destination [17] [11]. These schemes also return information on the level of reliability for each path. Based on this information, K paths are chosen and partitioned M streams of data are assigned to selected paths. A special case in MMDR is when $K < M$ during path assignment phase. In such a scenario, packets from multiple streams may share the same path and intermediate nodes might need information on significance of packets. Thus, any protocol designed based on MMDR should build that capacity within the packet streams. We restrict our discussion here for cases when $K \geq M$.

For the example H.264 data, since intra-coded I frames contain greater amount of data than their P counterparts, they may require further splitting over further paths for better load balancing, though, P frame streams may also requiring be split, depending on number of paths available for reaching back to destination and group of pictures (GOP) size used for encoding video. Intelligent data partitioning schemes are thus required that partition the frames at their boundaries such that despite losses in datasets for one frame other frames remain unaffected. For the scope of this work, we assume that is the case.

8.3.3 Distributed Progressive Error Recovery

We employ distributed progressive error recovery paradigm proposed in chapter 5 in conjunction with proposed MMDR framework.

8.3.4 Unequal Error Protection

The proposed architecture can be optionally combined with Unequal Error Protection (UEP), where, the channel coding rate is varied based on various node and network parameters. This does not come without few challenges though. In most of UEP literature, the packet size n is kept constant, but information bits k are varied based on [39]. Again, there are M streams and $K \leq M$ channel codes maybe used to encode them though a related issue is, for each variation in information bits, we may need a different channel code to encode it. Thus, since sensor networks are randomly deployed, tradeoff in sensor networks is greater amount of storage for each LDPC

code that might be required at each node depending on type of code used to encode that particular stream.

8.4 Simulation Setup

In most of the prevalent video coding architectures, video encoding is the primary computationally extensive task with complexity dominated by the motion-search operation. For the scope of this work, we assume that sensors have the capability to encode video though, efficient source coding of video still remains a wider research problem. In addition, we further assume that we are not facing motion estimation complexity issues that maybe encountered in coding over video sensor nodes and confine our focus towards data reliability aspects using progressive error recovery within the network. We consider a sensor node transmitting to a base station with multiple paths leading to the base station obtained through some underlying routing scheme such as directed diffusion [17]. We keep the assumptions regarding sensor nodes and the LDPC codes same as chapter 7.

The total energy for end-to-end transmission is taken as the transmission cost in transmitting \mathbf{n} bits over end-to-end multi-hop path, whereas, energy estimate for a single path in MMDR involves both transmit and computation energy for in-network processing. We set number of paths $K = 4$ with multiple realizations of the multi-hop network for each run such that average end-to-end equivalent error probability ϵ_{av} remains constant. All the results are averaged over 100 runs. We use two paths in order to send packets belonging to I frame and two for P frame. We set GOP size

Table 8.1. H.264 Encoder Parameters

Parameter	Value
Sequence	Foreman.qcif
YUV Format	4:2:0
QP	28
Profile IDC	High
Entropy Coding	CABAC
Sequence Type	IPP
Error Concealment	On

equal to 6. Both I & P-frame data is equally divided on each corresponding path such that despite losses in datasets for one frame other frames remain unaffected. On receive side, a resequencer collects all the packets, and assigns them to corresponding streams, which are then fed to channel and source decoding modules to obtain the reconstructed video (8.1).

8.5 Results

In this section, we present some preliminary simulation results that illustrate the efficiency of the proposed MMDR architecture in WSNs with use of an H.264 video encoder [2].

Figure 8.2 shows the performance of MMDR framework as compared to end-to-end channel coding. We see that MMDR outperforms end-to-end channel coding by significant margin, achieving reliable communication with much higher energy efficiency for average end-to-end equivalent error probabilities as high as $\epsilon_{av} = 0.125$ with a pair budget of $\Gamma(\bar{l})_{pair} = 8$.

Figure 8.3 shows the transmitted and reconstructed frame for both end-to-end channel coding and MMDR framework under different channel conditions with different pair-wise iteration budgets. The results clearly indicate much improved performance under MMDR for same energy levels used during end-to-end transmission.

Figure 8.4 uses peak signal-to-noise ratio (PSNR) in order to measure quality of video sequences for different end-to-end equivalent channel error probabilities. For a channel coding rate $R=0.5$ and average end-to-end equivalent error probability as high as 0.14, the plots clearly establish that MMDR outperforms end-to-end channel coding by considerable margins.

The results indicate that MMDR provides reliability to video streams in WSNs without fast depleting the energy resources of video sensors. They further depict that little bit of processing at intermediate nodes can greatly enhance video quality at the destination node. Use of path diversity gives the added flexibility in terms of added lifetime for WSNs. The rate-distortion curves obtained in Figure 8.2 and Figure 8.4 provide insight onto fine tuning the functional demands from the network in terms of adjusting the expected bit error rate/PSNR at the destination node for amount of energy spent in transmitting the bits reliably over end-to-end path.

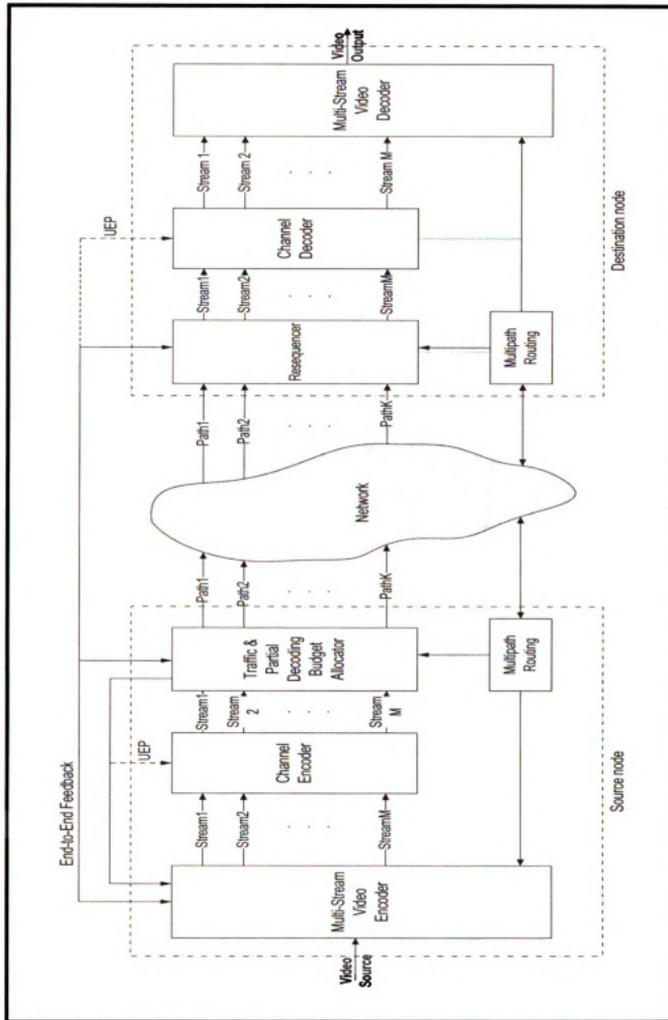


Figure 8.1. Multi-Path Multi-Stream reliability architecture for video transmission in Wireless Sensor Networks

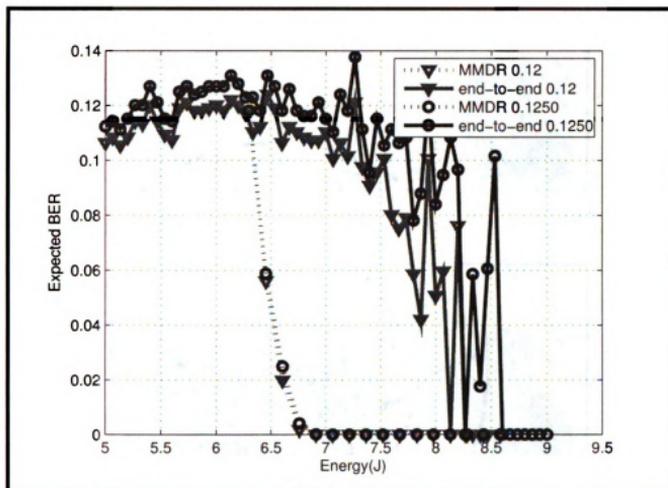


Figure 8.2. MMDR vs end-to-end transmission with $R = 0.5, K = 4, N = 4, \Gamma(\bar{l})_{pair} = 8$ a) $\epsilon_{av} = 0.12$ b) $\epsilon_{av} = 0.125$



(a) i- No losses ii- With 10% losses ($\epsilon_{av} = 0.10$) and end-2-end scheme iii- With $\epsilon_{av} = 0.10$ & MMDR scheme and $\Gamma(\bar{l})_{pair} = 6$



(b) i- No losses ii- With 11.75% losses ($\epsilon_{av} = 0.1175$) and end-2-end scheme iii- With 11.75% losses & D-PERA scheme with $\Gamma(\bar{l})_{pair} = 8$

Figure 8.3. Frame 2(P) for Foreman QCIF sequence

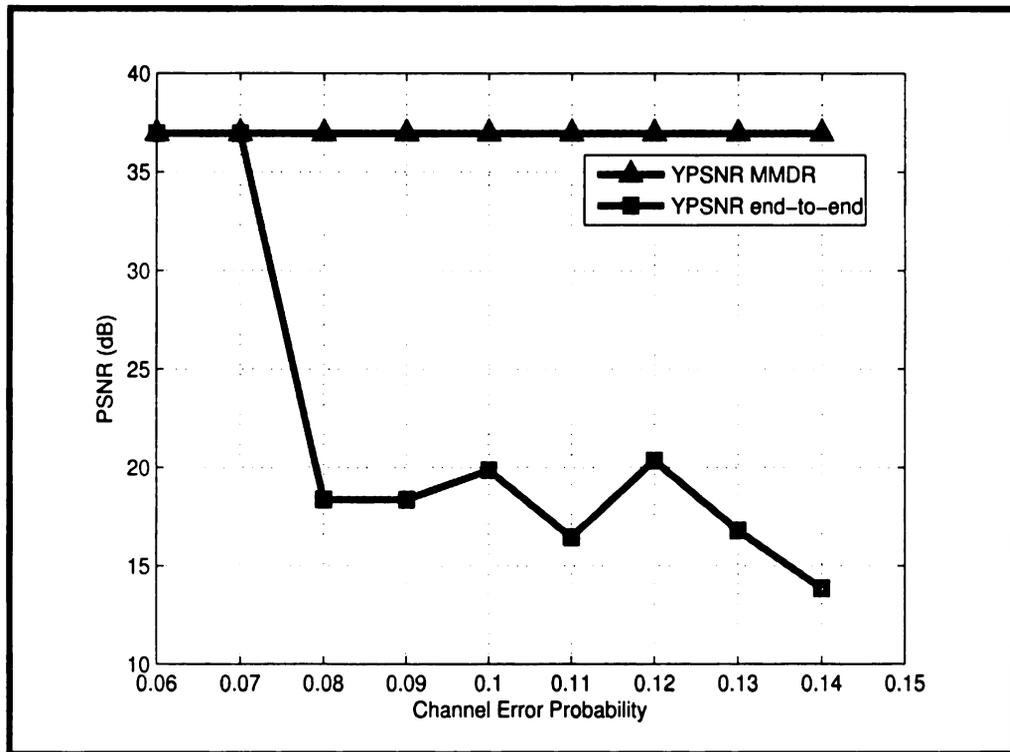


Figure 8.4. Average YPSNR for both MMDR with $\Gamma(\bar{l})_{pair} = 10$ and End-to-End for $R = 0.5$ LDPC encoded H.264 video.

CHAPTER 9

Conclusion and Future Work

In this work, we have presented a progressive error recovery paradigm that provides reliability to sensor network data in a progressive fashion. We have presented a framework that significantly enhances the decoding reliability at the receiver at the cost of processing within the network through partial decoding. We present both centralized and distributed approaches in order to map the decoding iterations to WSN for an iteratively decodable channel code.

The centralized approach presented optimally maps the decoding iterations to the intermediate nodes and performs significantly better than random assignment. We give bounds on the performance of the algorithm for computational energy spent within the network for partial decoding of ensembles of LDPC codes. In addition, we have proposed a methodological approach for wireless sensor network resilience against channel induced errors that ensures fairness to individual sensor nodes through rate adaptivity and introduction of bounds on processing at individual nodes. We give performance curves for variation in throughput at destination node using systematic puncturing of LDPC codes and discuss associated energy/throughput tradeoffs.

In addition, we have presented a distributed error recovery algorithm that ensures complete distribution of all the iterations to intermediate nodes. The bit error rates thus achieved outperform end-to-end channel coding by considerable margin for the amount of energy spent within the network. The setup obtained requires minimal exchange of messages needing only two extra transmissions per pair of nodes (follower node error rate and iteration assignment). We make no assumption regarding complete topological knowledge for the end to end path. The methodology thus achieved ensures pair-wise asynchrony in iteration assignment. The proposed proportional

warm-start provides fast convergence and guarantees minimal algorithmic overhead. Given a statistical model for variation of bit error rate with decoding iterations, our scheme is applicable to any iteratively decodable code.

Further, we have presented a multi-path distributed reliability scheme for wireless sensor networks. It provides a novel methodology to provide reliability to a Wireless Sensor Network, in a distributed fashion, in scenarios when there is asymmetry in the significance of data. The scheme is particularly useful in network conditions when high levels of end-to-end corruption are expected in the data sent over the multi-hop network. We show that how little bit of processing within the network carried out in an intelligent fashion can tremendously help in ensuring reliability of data at the destination.

We have shown the effectiveness of the proposed architecture with the help of relevant simulations using JPEG2000 images with LDPC codes. The proposed framework provides significantly better performance than prevalent end-to-end reliability schemes.

Further motivated by the stringent requirements due to resource constrained nature of sensor nodes, we have proposed an architecture to provide reliability to video in WSNs. We have described the architectural platform, theoretical foundations, as well as the bridge from theory to video practice, and presented promising experimental results based on real-world video sequences that establish the efficacy of our proposed solution over prevalent reliability schemes in WSNs.

The proposed framework can be extended to other prevalent video coding techniques including but not limited to DVC, SVC or MDC. As indicated in figure 8.1, a

future extension of this work can be when the packets are unequally protected using different LDPC codes. Similarly, adaptation of the framework with distributed video coding is another area that can be investigated.

Appendix: A

Table 1. Coefficients $\alpha(\epsilon_{ind}), \beta(\epsilon_{ind}), \gamma(\epsilon_{ind}), \psi(\epsilon_{ind})$, $\epsilon_{min} = 0.01$, $\epsilon_{max} = 0.08$, $l_{max} = 150$, $R = 0.50$, $\lambda(\theta) = 0.207\theta^6 + 0.271\theta^2 + 0.522\theta^1$

ϵ_{ind}	$\alpha(\epsilon_{ind})$	$\beta(\epsilon_{ind})$	$\gamma(\epsilon_{ind})$	$\psi(\epsilon_{ind})$
1	0.000000	0.000000	0.000000	0.000000
2	0.000000	0.000000	0.000000	0.000000
3	0.000000	0.000000	0.000000	0.000000
4	0.000000	0.000000	0.000000	0.000000
5	-0.77040	-0.49120	0.788700	-0.49920
6	-1.04800	-0.57960	1.071000	-0.58730
7	-0.95840	-0.49130	0.981200	-0.49900
8	-3.98e-18	6.07400	0.062490	-1.38600
9	-0.00036	-0.11460	0.046340	-1.13800
10	-7.110e-8	1.716000	0.046970	-1.04200
11	-1.346e-5	0.576800	0.076540	-1.27000
12	0.058820	-1.11000	0.005659	-0.29750
13	0.414500	-3.70900	0.035520	-0.56830
14	0.414500	-3.70900	0.035520	-0.56830
15	0.072770	-0.73810	-0.00013	0.194400
16	0.097310	-0.73360	-0.05287	-1.71600
17	-1.315e-7	1.419000	0.068030	-0.50910
18	0.160900	-0.77600	-0.49400	-2.83200
19	0.086340	-0.44410	-0.00198	0.142800
20	-1.70300	-0.27440	1.791000	-0.28160
21	-0.00019	0.225600	0.094360	-0.37780
22	0.020100	-0.33230	0.078730	-0.32970
23	-5.860e-6	0.439700	0.094610	-0.24380
24	-9.053e-7	0.577800	0.098810	-0.22640
25	-0.004318	0.054790	0.095930	-0.13270
26	-0.003394	0.051480	0.089930	-0.09533
27	-1.125e-5	0.235300	0.089780	-0.11180
28	-0.04832	-0.07443	0.123900	-0.07443
29	-0.43230	-0.08299	0.519000	-0.08480

Table 2. Coefficients $\alpha(\epsilon_{ind}), \beta(\epsilon_{ind}), \gamma(\epsilon_{ind}), \psi(\epsilon_{ind})$, $\epsilon_{min} = 0.01$, $\epsilon_{max} = 0.08$, $l_{max} = 150$, $R = 0.51$, $\lambda(\theta) = 0.207\theta^6 + 0.271\theta^2 + 0.522\theta^1$

ϵ_{ind}	$\alpha(\epsilon_{ind})$	$\beta(\epsilon_{ind})$	$\gamma(\epsilon_{ind})$	$\psi(\epsilon_{ind})$
1	0.000000	0.000000	0.000000	0.000000
2	0.000000	0.000000	0.000000	0.000000
3	0.000000	0.000000	0.000000	0.000000
4	-0.91380	-0.42110	0.936800	-0.42890
5	-1.77200	-0.65740	1.809000	-0.66450
6	-1.30600	-0.51590	1.337000	-0.52340
7	-1.61900	-0.58400	1.654000	-0.59110
8	-1.85300	-0.63560	1.901000	-0.64230
9	0.044220	-1.15700	0.020120	-0.98420
10	-1.21-10	2.061000	0.055540	-0.82360
11	-0.00012	0.128900	0.062320	-0.74030
12	-1.200000	-0.33280	1.253000	-0.33960
13	-1.34900	-0.38230	1.404000	-0.38900
14	-1.26200	-0.34670	1.316000	-0.35340
15	-1.24700	-0.28500	1.305000	-0.29170
16	-0.98590	-0.23840	1.047000	-0.24550
17	-4.074e-6	0.659300	0.064800	-0.36950
18	-0.00111	0.039210	0.070600	-0.34820
19	-5.808e-8	0.835800	0.077180	-0.38230
20	-1.184e-5	0.256500	0.082110	-0.38260
21	-3.646e-5	0.231600	0.072100	-0.18360
22	0.009536	-0.14580	0.063390	-0.14360
23	-0.04482	-0.11370	0.123500	-0.11360
24	0.076050	-0.39250	0.031780	0.000994
25	0.047450	-0.00811	0.002011	0.019310
26	0.071720	-0.51340	0.045690	0.000358
27	0.069430	-0.63150	0.053900	-3.098e-5
28	0.01408	-0.03402	0.051920	0.000232
29	0.04448	-0.44580	0.060100	4.565e-5

Table 3. Coefficients $\alpha(\epsilon_{ind}), \beta(\epsilon_{ind}), \gamma(\epsilon_{ind}), \psi(\epsilon_{ind})$, $\epsilon_{min} = 0.01$, $\epsilon_{max} = 0.08$, $l_{max} = 150$, $R = 0.52$, $\lambda(\theta) = 0.207\theta^6 + 0.271\theta^2 + 0.522\theta^1$

ϵ_{ind}	$\alpha(\epsilon_{ind})$	$\beta(\epsilon_{ind})$	$\gamma(\epsilon_{ind})$	$\psi(\epsilon_{ind})$
1	-0.00095	-0.14510	0.021970	-0.75990
2	-0.00421	-0.33890	0.032100	-0.75180
3	0.014430	-1.29300	0.052580	-1.43700
4	-199.1	-1.19800	199.2000	-1.19800
5	0.199200	-2.28500	0.007481	-0.37550
6	0.073890	-1.02700	-5.693e-6	0.533800
7	0.083440	-1.07900	-5.137e-7	0.934800
8	0.071620	-0.76590	-5.554e-6	0.605700
9	0.064960	-0.60780	-1.651e-5	0.618400
10	0.475000	-0.78070	-0.45020	-0.91870
11	0.146000	-0.58710	-0.11250	-0.99370
12	0.146000	-0.58710	-0.11250	-0.99370
13	0.142300	-0.58830	-0.10570	-1.01100
14	-0.84500	-0.09831	0.903800	-0.10580
15	-0.72450	-0.08762	0.784700	-0.09519
16	-0.00536	0.045930	0.076600	-0.19390
17	0.142400	-0.04939	0.213700	-0.08161
18	-0.00010	0.326100	0.079070	-0.17040
19	-0.02459	-0.01793	0.098860	-0.08698
20	-5.471e-5	0.151500	0.072670	-0.05144
21	0.027500	0.000000	0.027500	0.000000
22	0.028750	0.000000	0.028750	0.000000
23	0.030000	0.000000	0.030000	0.000000
24	0.031250	0.000000	0.031250	0.000000
25	0.032500	0.000000	0.032500	0.000000
26	0.033750	0.000000	0.033750	0.000000
27	0.035000	0.000000	0.035000	0.000000
28	0.037500	0.000000	0.037500	0.000000
29	0.040000	0.000000	0.040000	0.000000

Table 4. Coefficients $\alpha(\epsilon_{ind}), \beta(\epsilon_{ind}), \gamma(\epsilon_{ind}), \psi(\epsilon_{ind})$ $\epsilon_{min} = 0.01, \epsilon_{max} = 0.08,$
 $l_{max} = 150, R = 0.5, \text{PEG (3,6) Regular LDPC}$

ϵ_{ind}	$\alpha(\epsilon_{ind})$	$\beta(\epsilon_{ind})$	$\gamma(\epsilon_{ind})$	$\psi(\epsilon_{ind})$
1	0.000000	0.000000	0.000000	0.000000
2	0.000000	0.000000	0.000000	0.000000
3	0.000000	0.000000	0.000000	0.000000
4	-0.69460	-0.37620	0.712600	-0.38410
5	-0.97150	-0.47110	0.998500	-0.47830
6	-1.17000	-0.48130	1.206000	-0.48830
7	-1.12600	-0.41540	1.167000	-0.42230
8	0.31130	-1.09000	-0.30790	-1.37700
9	0.175600	-0.92100	-0.26610	-1.96300
10	0.435400	-0.66820	-0.38940	-0.74800
11	0.050130	-0.36520	0.017250	-0.36810
12	0.068400	-0.27110	0.000198	0.000539
13	0.009578	-0.04029	0.063590	-0.24560
14	0.065780	-0.14410	0.005974	-0.00118
15	0.054300	-0.14570	0.022660	-0.00015

Table 5. Coefficients $\alpha(\epsilon_{ind}), \beta(\epsilon_{ind}), \gamma(\epsilon_{ind}), \psi(\epsilon_{ind})$, $\epsilon_{min} = 0.005$, $\epsilon_{max} = 0.08$, $l_{max} = 150$, $R = 0.569$, $\lambda(\theta) = 0.207\theta^6 + 0.271\theta^2 + 0.522\theta^1$

ϵ_{ind}	$\alpha(\epsilon_{ind})$	$\beta(\epsilon_{ind})$	$\gamma(\epsilon_{ind})$	$\psi(\epsilon_{ind})$
1	-0.020270	-1.924000	0.027520	-2.060000
2	0.023630	-2.398000	0.027520	0.000000
3	-0.235400	-0.318700	0.000000	-0.327400
4	-0.000398	-0.491000	0.242800	-1.441000
5	0.000458	-0.381900	0.034490	-1.272000
6	0.045160	-1.050000	0.039560	0.000000
7	0.046460	-0.82750	0.000000	-0.286900
8	0.001199	-0.322400	0.051330	-0.735200
9	0.000270	-0.151600	0.055600	-0.572200
10	0.057380	-0.429300	1.033e-14	0.157400
11	0.114100	-0.311200	-0.051560	-0.282500
12	0.064360	-0.271100	0.001997	-0.042900
13	0.061870	-0.196400	0.003563	-0.000210
14	0.055990	-0.160700	0.012830	-0.000351
15	0.040590	-0.139000	0.029850	-0.000183
16	0.029480	-0.221600	0.049870	-4.751e-5

Bibliography

BIBLIOGRAPHY

- [1] *Crossbow Imote2 Multimedia Board IMB 400 for Wireless Sensor Networks, Product File*, 2008.
- [2] Jm reference software jvt-ad010, 2008.
- [3] I.F. Akyildiz, T. Melodia, and K.R. Chowdhury. A survey on wireless multimedia sensor networks. *Computer Networks*, 51(4):921–960, 2007.
- [4] IF Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazines*, 40(8):102–114, 2002.
- [5] J. Barros and S.D. Servetto. The sensor reachback problem. *submitted to the IEEE Trans. Inform. Theory*.
- [6] L. Bazzi, TJ Richardson, and RL Urbanke. Exact thresholds and optimal codes for the binary-symmetric channel and Gallager’s decoding algorithm A. *IEEE Transactions on Information Theory*, 50(9):2010–2021, 2004.
- [7] S.P. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [8] T.M. Cover, J.A. Thomas, J. Wiley, and W. InterScience. Elements of information theory. 1991.
- [9] MPC Fossorier, M. Mihaljevic, and H. Imai. Reduced complexity iterative decoding of low-density parity checkcodes based on belief propagation. *IEEE Transactions on Communications*, 47(5):673–680, 1999.
- [10] R. Gallager. Low-density parity-check codes. *IEEE Transactions on Information Theory*, 8(1):21–28, 1962.
- [11] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(4):11–25, 2001.
- [12] Ramesh Govindan. Ieee comsoc wireless sensor networks tutorial. <http://www.comsoc.org/freetutorials/nsc/>, 2006.

- [13] VK Goyal. Multiple description coding: Compression meets the network. *IEEE Signal Processing Magazine*, 18(5):74–93, 2001.
- [14] P. Gupta and PR Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, 2000.
- [15] X.Y. Hu, E. Eleftheriou, and DM Arnold. Regular and irregular progressive edge-growth tanner graphs. *IEEE Transactions on Information Theory*, 51(1):386–398, 2005.
- [16] Muhammad Usman Ilyas and Hayder Radha. Increasing network lifetime of an IEEE 802.15.4 wireless sensor network by energy efficient routing. In *Proceedings of IEEE International Conference on Communications (ICC 06)*, 2006.
- [17] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva. Directed diffusion for wireless sensor networking. *IEEE/ACM Transactions On Networking*, 11(1), 2003.
- [18] S. Madden, M.J. Franklin, J.M. Hellerstein, and W. Hong. TAG: a Tiny Aggregation Service for Ad-Hoc Sensor Networks. *Proceedings of the ACM Symposium on Operating System Design and Implementation (OSDI)*, 2002.
- [19] S. Mao, S. Lin, SS Panwar, Y. Wang, and E. Celebi. Video transport over ad hoc networks: Multistream coding with multipath transport. *IEEE Journal on Selected Areas in Communications*, 21(10):1721–1737, 2003.
- [20] R. Puri, A. Majumdar, P. Ishwar, K. Ramchandran, P.P. Inc, and CA San Jose. Distributed video coding in wireless sensor networks. *IEEE Signal Processing Magazine*, 23(4):94–106, 2006.
- [21] SB Qaisar, S. Karande, K. Misra, and H. Radha. Optimally Mapping an Iterative Channel Decoding Algorithm to a Wireless Sensor Network. *IEEE International Conference on Communications, ICC'07.*, pages 3283–3288, 2007.
- [22] SB Qaisar and H. Radha. Optimal Progressive Error Recovery for Wireless Sensor Networks using Irregular LDPC Codes. pages 232–237, 2007.
- [23] S.B. Qaisar and H. Radha. Low complexity distributed reliability for Wireless Sensor Networks. *Conference on Information Sciences and Systems, CISS'08.*, pages 647–652, 2008.
- [24] H. Radha, M. Vetterli, and R. Leonardi. Image compression using binary space partitioning trees. *IEEE Transactions on Image Processing*, 5(12):1610–1624, 1996.

- [25] HM Radha, M. Van der Schaar, and Y. Chen. The MPEG-4 fine-grained scalable video coding method for multimediasstreaming over IP. *IEEE Transactions on Multimedia*, 3(1):53–68, 2001.
- [26] T.S. Rappaport. *Wireless Communications: Principles and Practice*. IEEE Press Piscataway, NJ, USA, 1996.
- [27] T. Richardson, A. Shokrollahi, and R. Urbanke. Design of provably good low-density parity check codes. *IEEE Trans. Inform. Theory*, 47:619–637, 2001.
- [28] TJ Richardson, MA Shokrollahi, and RL Urbanke. Design of capacity-approaching irregular low-density parity-checkcodes. *IEEE Transactions on Information Theory*, 47(2):619–637, 2001.
- [29] T.J. Richardson and R.L. Urbanke. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Transactions on Information Theory*, 47(2):599–618, 2001.
- [30] Y. Sankarasubramaniam, IF Akyildiz, and SW McLaughlin. Energy efficiency based packet size optimization in wireless sensor networks. *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications*, pages 1–8, 2003.
- [31] M. Sartipi and F. Fekri. Source and channel coding in wireless sensor networks using LDPC codes. *First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, pages 309–316, 2004.
- [32] I. Sason and R. Urbanke. Parity-check density versus performance of binary linear block codes over memoryless symmetric channels. *IEEE Transactions on Information Theory*, 49(7):1611–1635, 2003.
- [33] G.M. Schuster and A.K. Katsaggelos. *Rate-Distortion Based Video Compression: Optimal Video Frame Compression and Object Boundary Encoding*. Kluwer Academic Publishers, 1997.
- [34] RC Shah and JM Rabaey. Energy aware routing for low energy ad hoc sensor networks. 1, 2002.
- [35] E. Shih, S.H. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan. Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks. *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 272–287, 2001.

- [36] G.J. Sullivan and T. Wiegand. Rate-distortion optimization for video compression. *IEEE Signal Processing Magazine*, 15(6):74–90, 1998.
- [37] D.S. Taubman, M.W. Marcellin, and M. Rabbani. JPEG2000: Image compression fundamentals, standards and practice. *Journal of Electronic Imaging*, 11:286, 2002.
- [38] D. Tuninetti and C. Fragouli. Processing along the way: forwarding vs. coding. *ISITA 2005*.
- [39] M. van der Schaar and H. Radha. Unequal packet loss resilience for fine-granular-scalability video. *IEEE Transactions on Multimedia*, 3(4):381–394, 2001.
- [40] Y. Wang, A.R. Reibman, and S. Lin. Multiple description coding for video delivery. *Proceedings of the IEEE*, 93(1):57–70, 2005.
- [41] M. Wu, S.S. Karande, and H. Radha. Network-embedded FEC for optimum throughput of multicast packet video. *Signal Processing: Image Communication*, 20(8):728–742, 2005.
- [42] H. Wymeersch, H. Steendam, and M. Moeneclaey. Log-domain decoding of LDPC codes over GF (q). *IEEE International Conference on Communications*, 2004.

MICHIGAN STATE UNIVERSITY LIBRARIES



3 1293 03063 1000