Distributed Diversity in Hybrid Wireless Networks

A dissertation presented
by

Yin Wang

to the Faculty of the Graduate School
of the College of Computer and Information Science
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Northeastern University
Boston, Massachusetts

September, 2010
Abstract

Wireless communication networks span a wide range of applications such as traditional telephony, internet data access, and cyber-physical infrastructure monitoring. However, the quality of service is still lacking, and problems such as dropped calls, dead spots, and slow network speed are still very common. Those are due to fundamental constraints such as the scarcity of radio-frequency spectrum, signal propagation effects, e.g., channel fading and shadowing, and the small form factor of mobile devices with limited energy capacity and antenna diversity.

In this thesis, we explore a new cooperative communication model. We investigate communication strategies that exploit the channel and traffic diversity across a set of cooperating mobile nodes equipped with multiple radio frequency interfaces (e.g., cellular, WiFi). The research aims at developing a framework for cooperation across multiple network layers to improve robustness, throughput, and delays. The proposed cooperation is based on two strategies: signal Combining and traffic-multiplexing. Within this framework, we propose and evaluate several distributed cooperation techniques operating at different hierarchical levels with resource constraints such as short-range RF bandwidth. Our evaluation is based on a combination of analysis, simulations, and real world experiments using system prototypes.
I would like to sincerely thank my advisor, Guevara Noubir, for his long time support of my research as well as serving on my thesis committee. His insight, guidance and inspiration are invaluable and will effect me beyond school. I would also like to thank the rest of my thesis committee: Rajmohan Rajaraman, Ravi Sundaram and Benyuan Liu for their suggestions on my research and reviewing my thesis. I would also like to thank Bishal Thapa for his careful proofreading and constructive comments. I am also grateful to my academic mentor Anges Chan for her generous support of my study abroad at the beginning years and introducing me to the research field. I would also like to appreciate the support from my family in China. Finally, I would like to take this opportunity to express my gratitude to my friends across the world for their encouragement and support of my life.
# Contents

Abstract i  
Acknowledgments iii  
Contents v  
List of Figures ix  

1 Introduction 1  
1.1 Overview and Motivation 1  
1.2 Distributed Diversity Framework 5  
  1.2.1 Models 7  
  1.2.2 Diversity 11  
  1.2.3 Cooperation Approaches 13  
  1.2.4 Modes of Operation 15  
  1.2.5 System Constraints 15  
  1.2.6 Security and Fairness 17  
1.3 Evaluation and Experimentation 17  
  1.3.1 Metrics 17  
  1.3.2 Testbeds and Developed Prototypes 20  
1.4 Contributions 20
1.5 Related Work ............................................... 21

2 Distributed Hierarchical Combining Strategy .......... 25
  2.1 Motivation and Technical Background ................. 26
    2.1.1 Motivation for Distributed Diversity ............... 27
    2.1.2 System Model ...................................... 28
    2.1.3 Maximum Ratio Combining .......................... 31
  2.2 Hierarchical Priority Combining ....................... 33
    2.2.1 The Hierarchical Structure ....................... 33
    2.2.2 Two-Phase Protocol Design ....................... 35

3 Threshold Maximum Ratio Combining .................... 39
  3.1 Introduction ............................................ 39
  3.2 Probability Distribution of SNR for TMRC .......... 41
  3.3 Performance Analysis ................................ 45
    3.3.1 Outage Probability ................................. 46
    3.3.2 Bit Error Rate .................................... 47
    3.3.3 Throughput ........................................ 48
    3.3.4 Local Bandwidth Usage ............................ 50
  3.4 Protocol Implementation of TMRC ..................... 51
  3.5 Generalization of TMRC ............................... 55
  3.6 Summary on TMRC and RMRC ............................ 56

4 Priority Maximum Ratio Combining ..................... 59
  4.1 Introduction ............................................ 59
  4.2 SPMRC: SNR Distribution for $N = 1, 2, 3$ .......... 61
  4.3 PMRC: SNR Distribution for $N = 2, 3, 4$ ............ 64
  4.4 Performance Analysis ................................ 65
    4.4.1 Outage Probability ............................... 66
4.4.2 Bit Error Rate ........................................... 68
4.4.3 Throughput ............................................... 70
4.4.4 Local Bandwidth Usage ............................... 71
4.5 Protocol Implementation for PMRC ................... 73
4.6 Simulations .................................................. 75

5 Post Soft-Demodulation Combining .................... 77
  5.1 Introduction .............................................. 77
  5.2 Maximum Likelihood Soft Combining ................. 79
  5.3 SNR Probability Distribution for Priority Signal Source Vector 82
  5.4 Performance Analysis .................................. 84
    5.4.1 Bit Error Rate .................................... 84
    5.4.2 Throughput ....................................... 86
    5.4.3 Local Bandwidth Usage .......................... 88
  5.5 Experiments for Post Soft-Demodulation Combining .... 89
    5.5.1 GNU Radio and USRP Testbed .................... 89
    5.5.2 Experimentation Results .......................... 91

6 Distributed Traffic Multiplexing ....................... 97
  6.1 Background and Motivation ........................... 97
  6.2 Model and Approach ................................... 99
  6.3 Stationary Regime for Blocked-Poisson Traffic, and Exponential Service Time 101
  6.4 Traffic Pattern and Simulation ....................... 106
    6.4.1 Blocked-Poisson Traffic (Serial vs. Parallel-Multiplexing) 106
    6.4.2 Heavy-Tail Traffic ............................... 108
  6.5 Rate Allocation ......................................... 110

7 Distributed Link Bonding Prototyping ................ 113
<table>
<thead>
<tr>
<th>CONTENTS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1 Cooperative Solutions</td>
<td>113</td>
</tr>
<tr>
<td>7.2 Tunnelling Based Architecture and Prototype</td>
<td>118</td>
</tr>
<tr>
<td>7.3 Neptune Architecture and Prototype</td>
<td>121</td>
</tr>
<tr>
<td>7.4 Experimentation</td>
<td>124</td>
</tr>
<tr>
<td>8 Conclusion</td>
<td>131</td>
</tr>
<tr>
<td>Bibliography</td>
<td>133</td>
</tr>
</tbody>
</table>
List of Figures

1.1 Distributed diversity scenario ........................................ 3
1.2 Cooperation at multiple network layers. ............................... 14
1.3 Modes of operation. .......................................................... 16

2.1 Maximum Ratio Combining diagram ................................. 32

3.1 PDF of $\gamma$ of each branch at master side in TMRC ................. 41
3.2 The outage prob. of the master node in TMRC ($\gamma_T = \gamma_0$) with $M = 2, 3, 4, 5$ compared to non cooperative case $M = 1$. .... 46
3.3 The outage prob. of the master node with different $\gamma_T$ in TMRC compared to MRC and non cooperative $M = 1$. ................. 46
3.4 Bit Error Rate of coherent MSK demodulator under TMRC with different $\gamma_T$ compared to MRC and non-cooperative case ($M = 1$). ......................................................... 48
3.5 Bit Error Rate of coherent MSK demodulator under TMRC ($\gamma_T = 5dB$) with $M = 2, 3, 4, 5$ and non-cooperative case ($M = 1$). 48
3.6 Throughput of TMRC with various $E_b/N_0$(dB) and the number of nodes in cooperation. .................................................. 49
3.7 Local bandwidth consumption under different $E_b/N_0$ in TMRC ($M = 5$). ......................................................... 50
3.8 Bit error rate vs local bandwidth usage in TMRC ($\gamma_T = 5dB, M = 5$). .................................................. 50
3.9 Percentages of the energy gain and the bandwidth requirement of TMRC over MRC ($M = 5$). ............................... 51
3.10 A running instant of TMRC with 1 master node and 3 assisting nodes. (I) The long-range TDMA channel (II) The short-range channel. ................................................................. 52
3.11 Various $TX(\gamma)$ functions for RMRC. .................................................. 56
4.1 Outage Probability of PMRC ($N = 2, ..., 4$) vs. MRC and non-cooperative case. .................................................. 67
4.2 Impact of $M$ on the performance of PMRC. ............................... 67
4.3 Bit Error Rate of coherent MSK demodulator under PMRC ($N = 2, ..., 4$) vs. MRC and non-cooperative case. ................. 68
4.4 Impact of $M$ on the performance of PMRC in terms of Bit Error Rate. .................................................. 68
4.5 Bit error rate: PMRC vs TMRC ($M=5$). .................................................. 69
4.6 Local bandwidth usage of (5, 2)-PMRC and TMRC ($\gamma_T = 0dB, M = 5$). .................................................. 69
4.7 Throughput of PMRC with various $E_b/N_0$(dB) and the number of nodes in cooperation. .................................................. 70
4.8 Bit error rate and local bandwidth usage of (5, 4)-PMRC based HPC. (Packet Length is 1500 bytes) ............................... 72
4.9 BER performance of PMRC with pilot-based channel estimation. ............................... 75
5.1 Bit Error Rate of coherent MSK demodulator under PSDC ($N = 2, ..., 4$) vs. MRC and Non-Cooperation. ............................... 85
5.2 Impact of $M$ on the performance of Post-Soft Demodulation Combining in terms of Bit Error Rate. .................................................. 85
5.3 Bit Error Rate comparison between PSDC vs TMRC $(M=5)$. .... 86
5.4 Bit Error Rate comparison between PSDC vs PMRC. ................. 86
5.5 Throughput in different PSDC cooperation scenarios and $E_b/N_0$(dB) 87
5.6 Throughput comparison for PSDC and PMRC $(M=3)$. ............... 88
5.7 Throughput comparison for PSDC and PMRC $(M=5)$. ............... 88
5.8 USRP Block Diagram .............................................................. 90
5.9 Bit Error Rate of PSDC by experiments $(N = 2)$. ...................... 92
5.10 Bit Error Rate of PSDC by experiments $(M = 5)$. ..................... 92
5.11 Throughput of PSDC by experiments ....................................... 92
5.12 The distribution of the soft-decision values from the receiver if zeros are sent out. ................................................................. 94
5.13 Bit Error Rate of PSDC theory vs. experiments ......................... 94
5.14 Bit Error Rate of PSDC at different quantization levels. $(M = 5, N = 3)$ ................................................................. 94
6.1 Markov chain for each node in non-cooperative mode ............... 101
6.2 Markov chain for $m$ cooperating nodes in serial-multiplexing mode. 103
6.3 Average delay per packet for serial and parallel-multiplexing with Blocked-Poisson traffic. ..................................................... 107
6.4 Perceived Throughput as a function of load for Blocked-Poisson traffic and $m = 1, \cdots, 5$. ..................................................... 108
6.5 Perceived throughput as a function of load for heavy-tail traffic and $m = 1, \cdots, 5$. ..................................................... 108
6.6 Ratio of perceived throughput of parallel to serial-multiplexing for Blocked-Poisson traffic. ..................................................... 109
6.7 Ratio of perceived throughput of parallel to serial-multiplexing for heavy-tail traffic. ........................................ 109
6.8 Comparison of perceived throughput under exponential and heavy-tail traffic for serial-multiplexing. Both traffics have the same average request length. ........................................ 110
6.9 Ratio of perceived throughput for heavy-tail traffic to exponential using serial-multiplexing. For the same average request length, heavy-tail traffic results in substantially better throughput. ... 110

7.1 Configuration of the proxy based DLB platform. ............... 117
7.2 Neptune Agent controller on Google Android platform. ...... 124
7.3 A testing tool for DLB prototype. ................................. 124
7.4 Downloading throughput of Neptune DLB with two Verizon Droid phones in cooperation. ........................................ 128
7.5 Uploading throughput of Neptune DLB with two Verizon Droid phones in cooperation. ........................................ 128
7.6 Downloading throughput of Neptune DLB with one Verizon Droid and one Tmobile G1 in cooperation. ......................... 129
7.7 Uploading throughput of Neptune DLB with one Verizon Droid and one Tmobile G1 in cooperation. ......................... 129
7.8 Downloading throughput of Neptune DLB with one Verizon Droid and one AT&T iPhone 3G in cooperation. ............... 129
7.9 Uploading throughput of Neptune DLB with one Verizon Droid and one AT&T iPhone 3G in cooperation. ............... 129
CHAPTER 1

Introduction

1.1 Overview and Motivation

Wireless communication networks are enabling an ever increasing set of applications. This set spans domain areas such as traditional telephony, ubiquitous access to information and computation, cyber-physical infrastructure monitoring, and disaster recovery. According to a recent report from International Telecommunication Union (ITU) [69], there are 4.6 billion mobile phone subscribers in the world by the year of 2009. However, the wireless service quality and scalability are limited by fundamental constraints. These include a scarce radio-frequency spectrum, signal propagation effects such as fading and shadowing causing areas with limited coverage, and the small form factor of mobile devices with a limited capacity to store energy. The solution current operators opt for is to deploy additional base stations to shrink the cell size and thereby reduce the distance to the base stations [63]. However, this strategy is ineffective and costly. In this thesis, we propose to explore a new communication model, where multiple mobile nodes cooperate with each other and with the base stations to improve the service quality
and system performance. We will investigate communication strategies that exploit the channel and traffic diversity across a set of cooperating mobile nodes equipped with multiple radio interfaces (e.g., cellular and WiFi).

The networking architecture and software protocol stack of current wireless systems have fundamental limitations in addressing the unique characteristics of wireless communications such as shared medium, signal fading and shadowing, and interference. In the case of cellular networks, although various new techniques have been introduced in the third generation systems, and have been deployed for several years, quality of service is still lacking, and problems such as dropped calls, dead spots, and slow network speed are still very common. According to a recent survey [9], 55% of smartphone users are not satisfied with the network services and are experiencing a high percentage of dropped calls and slow mobile-access Internet speed (also reported in [4, 10]). Other reports show a dropped call rate as high as 30% in some urban areas [1, 2]. These problems constitute a critical research challenge because of the increased bandwidth demand by future applications.

Diversity and cooperation, as general mechanisms to increase the robustness and efficiency of wireless communication systems, have been studied for many years [60, 62, 34, 68], but very little research has been done on distributed cross-layer systems with multiple types of air-interfaces (e.g., GSM, HSPA, WiMAX and IEEE802.11) and considering the unique characteristics of each of the interfaces. With increased hardware integration, fast computation, and high user density, the cooperation between nearby devices is becoming possible and even necessary given the increased demand for bandwidth.
A scenario of distributed diversity and cooperation is depicted in Figure 1.1. In this environment, there are different types of cooperation possible and each exploits one or multiple diversity gains. For example, there are three mobile users each with a cellular phone. Due to obstructing objects and the distance to the base station, they suffer from the typical channel fading and path loss (attenuation) that impair urban cellular communication. With cooperation, the long-range cellular signals are (1) independently received at each of the three nodes, (2) relayed through the high speed local wireless network, and (3) combined at the destination node. This cooperation can significantly improve the Signal to Noise Ratio (SNR), Bit Error Rate (BER) and throughput. It leads to improved coverage and a system capacity boost. Furthermore, it reduces interference as the base stations do not have to increase their transmission power to overcome the channel fading in order to reach mobile nodes. Another example is when multiple users contend to access the web through cellular networks; most of the time, their long-range cellular links are not used, and even if they are simultaneously
browsing, the probability that two users are simultaneously downloading is small (because of the time users take to read and think about the downloaded content). Therefore, multiplexing the user traffic over a distributed bundled link can result in significant performance improvement.

The research work in this thesis aims at developing a framework for cooperation across multiple network stack layers to improve robustness, throughput, and delays. We focus on a specific heterogeneous network with devices equipped with two types of radio frequency (RF) interfaces: a short-range high data-rate interface (e.g., WiFi), and a long-range low data-rate interface (e.g., cellular). Within this framework, we propose and evaluate a set of distributed cooperation techniques operating at different hierarchical levels with various resource constraints. The proposed cooperation is based on two loosely-coupled strategies: signal combining and traffic multiplexing, which exploit two types of diversities: channel diversity and traffic diversity. The framework provides a dynamic way to adopt suitable cooperation techniques based on user’s requirement, the channel conditions and short-range RF bandwidth limitations. This research aims at answering several questions:

- What are the possible and practical diversity cooperation techniques that can be adopted to improve mobile wireless communications?

- How much performance improvement can we achieve and what are the constraints on each strategy?

- How can those strategies be implemented in real world testbeds? Are any of them feasible on the current hardware platforms?
1.2. DISTRIBUTED DIVERSITY FRAMEWORK

The rest of the thesis is organized as follows. In the rest of Chapter 1, we introduce the proposed framework, the considered models for the network, channel and traffic, the cooperation methods, constraints, and the evaluation methods. In Chapters 2, 3, 4 and 5, we propose several cooperation mechanisms and the protocols that exploit the channel diversity through signal combining. We investigate their performance analytically, and through simulations. Then, we demonstrate their potential improvement through experimentation using our implementation on a software-defined radio testbed. In Chapters 6 and 7, we propose cooperation mechanisms, a network architecture, and implementations to exploit traffic diversity through traffic multiplexing over heterogeneous networks. In addition to analytically characterizing the performance of the proposed mechanisms, we implement some of the proposed protocols, and demonstrate their capabilities within and across several cellular network operators.

1.2 Distributed Diversity Framework

We consider a hybrid network where the mobile nodes are equipped with two radio communication interfaces: a long-range low data-rate interface, and a short-range high data-rate interface. The framework aims at exploiting the gains from channel diversity and traffic diversity through two basic approaches: signal combining and traffic multiplexing.

The performance of long-range cellular links is limited by the shadowing and channel fading caused by multipath propagation and mobility. These are critical problems in mobile communication as they result in dead-signal areas and localized poor system performance. Channel diversity is a typi-
cal approach to mitigate those problems through independent transmission paths. Many existing technologies, such as multiple-input and multiple-output (MIMO), require multiple antennas to be co-located on the same device. But, due to the requirements of the minimum spatial separation and high cost of RF front ends, it is difficult to implement these schemes on a single small form factor device such as a cell phone [64]. Instead, our cooperation framework intends to take advantage of the RF front ends of a set of geographically distributed devices.

A limiting factor of the long-range cellular communication is its relatively low data-rate. Users of mobile phones usually make one request at a time. This results in an imbalanced traffic and underused bandwidth. However, neighbouring nodes could take advantage by pooling their links together through the high-speed short-range communication interface, and benefit through traffic multiplexing. This results in an improvement of the actual throughput experienced by users and the overall network utilization.

The cooperation strategies are designed to be transparent to applications; therefore, the existing applications would have an improved performance without requiring any awareness or modification. In the following, we outline the models, the key principle, our approaches, the design and the constraints of the proposed system.
1.2. DISTRIBUTED DIVERSITY FRAMEWORK

1.2.1 Models

1.2.1.1 Topology

In the proposed distributed diversity system, the nodes consist of a group of nearby mobile stations (MS), and base stations or base transceiver stations (BTS). The base stations are controlled by the base station controller (BSC), which dictates the carrier frequencies, communication power and rate, etc. The base stations are also connected to the backbone which leads to the telephone network and the Internet.

Communication between mobile stations and base stations is through long-range low data-rate links. Mobile stations can also communicate with each other through short-range high data-rate links. Because of the short distance, any mobile station can communicate with any other mobile station within one hop. For example, in Figure 1.1 a base station $BTS_1$ is communicating with a mobile station $MS_1$ and another mobile station $MS_2$ in the vicinity through long-range low data-rate links; the links from $MS_1$ to $MS_2$, from $MS_2$ to $MS_3$, and from $MS_3$ to $MS_1$ are short-range high data-rate links.

In our framework, due to the shared medium and short distance, the mobile stations are able to broadcast messages through the short-range air interface. Besides, they are also likely to be covered by the same base station, so all the long-range communications are broadcasts. Here, we assume each of the mobile stations can be set to receive other mobile stations’ long-range communications even if it is not associated to that base station, which is critical for signal combining to work. Due to the physical separation of the mobile stations, the long-range links such as $BTS_1$ to $MS_1$ and $BTS_1$ to
$M S_2$ are independent communication paths. We also assume that all of the mobile stations are connected to the Internet through their own cellular networks, which allows us to exploit the gain of traffic diversity.

### 1.2.1.2 Channel Models

In communication systems, *Additive White Gaussian Noise (AWGN)* is a commonly used channel model that only accounts for the white noise. Under this channel model, and for a specific modulation, the SNR determines the symbol error rate. In contrast, in wireless communications various types of fading cause the signal power to fluctuate over time and space due to multipath propagation and shadowing. This is the case of urban cellular communications, where the signal travels through multiple paths due to the reflection from objects such as buildings and trees. The signals from these paths might add up or cancel each other and result in weak signals. We recall several popular channel fading models. Our analysis in the following chapters is based on the Rayleigh channel, which models the fading that typically occurs in cellular communications [34].

- **Rayleigh Channel:** There is no dominant propagation path between the transmitter and the receiver, and multiple delayed signals from different paths add up at the receiver. It is usually the case where there is no line-of-sight (LOS). In this case, the in-phase and quadrature components of the received signal are both zero-mean Gaussian random variables, and the received signal envelope $r$ is Rayleigh distributed [62]. The probability density function for $r$ is given by

$$p(r) = \frac{r}{\sigma^2} \exp\left(\frac{-r^2}{2\sigma^2}\right), \quad r \geq 0,$$
where $\sigma^2$ is the variance, and represents the AC power in the signal envelope.

- **Rician Channel:** When there is a dominant fixed signal component present, such as LOS propagation path, the in-phase and quadrature components of the received signal are no longer zero-mean random variables. In this case, the received signal equals the superposition of a complex Gaussian component and a LOS component. It can be shown that the received signal envelope is Rician distributed [62]. The probability density function is given by

$$p(r) = \frac{r}{\sigma^2} \exp \left( -\frac{(r^2 + A^2)}{2\sigma^2} \right) I_0 \left( \frac{Ar}{\sigma^2} \right), \ A \geq 0, r \geq 0,$$

where $A$ is the peak amplitude of the dominant signal and $I_0$ is the modified Bessel function of zero-order. The Rician distribution is often described in terms of a fading parameter $K$, which measures the severity of the fading. It is defined by

$$K = \frac{A^2}{2\sigma^2}.$$

For $K = 0$, it is Rayleigh fading, where there is no LOS component. For $K = \infty$, it is the no fading case, where there is only a LOS component and no other propagation paths.

- **Nakagami Channel:** Nakagami fading distribution is a more general model to fit a variety of empirical measurements that do not fit into Rayleigh model or Rician model [34]. The probability density function is given by

$$p(r) = \frac{2m^m r^{2m-1}}{\Gamma(m) P_m} \exp \left( -\frac{m r^2}{P} \right), m \geq 0.5,$$
where $\Gamma$ is the Gamma function, and $\bar{P}$ is the average received power which is defined as

$$\bar{P} = \int_0^\infty r^2 \cdot p(r) dr.$$ 

For $m = 1$, it becomes Rayleigh fading. For $m = (K+1)^2/(2K+1)$, it is approximately Rician fading with a fading parameter $K$. For $m = \infty$, there is no fading.

### 1.2.1.3 Traffic Model

The application on the mobile nodes generates requests, and sends them to the remote application servers through their long-range cellular links. The responses such as web pages from the application servers are returned back to the mobile nodes. In the proposed Distributed Link-Bonding (DLB) system, we assume the long-range cellular links are limited by low speed at a certain data-rate.

We consider a Blocked-Poisson distribution for the traffic. The requests are generated according to a specific distribution such as Poisson and the node is blocked until the current request is fulfilled. This process is motivated by the fact that mobile phone users usually make one request at a time, e.g., browse a web page and wait until it is completely loaded before making another request.

The service time corresponds to the length of a web page, or the duration of a session. We consider two types of distributions: exponential and heavy-tail. The exponential distribution is commonly used to model the traffic in networks and for queues analysis, but the investigation of the Internet traffic has suggested that the request sizes or service durations on the Internet are
1.2. DISTRIBUTED DIVERSITY FRAMEWORK

often according to heavy-tail distributions [59, 25, 36]. The heavy-tailed traffic means that in terms of the size or duration of the requests that there are not only many short requests, but also a few very long requests. In contrast, the exponential traffic with the same mean would have far more medium size requests. In our simulations, we use Pareto distribution to model the heavy-tail traffic. Let $l$ be the random variable that represents the duration of the request. So according to Pareto distribution, the probability density function for $l$ is

$$p(l) = \frac{ab^a}{l^{a+1}}, \quad l \geq b,$$

where $a$ and $b$ are the parameters, which define the shape and scale of the distribution respectively. The distribution function is

$$F(l) = 1 - \left(\frac{b}{l}\right)^a, \quad x \geq b$$

The mean and variance are

$$\mu = \frac{ab}{a-1}$$

$$\sigma^2 = \frac{ab^2}{(a-1)^2(a-2)}$$

1.2.2 Diversity

Diversity is a powerful technique to combat communication errors caused by channel fading. It exploits the benefit from independent signal paths, and has no overhead in comparison to equalization, which uses training sequence. There are a variety of diversity techniques such as time diversity, frequency diversity, space diversity and traffic diversity. The techniques used in the proposed cooperative framework mainly involve space diversity and
traffic diversity. Below, we provide a short description of the most popular diversity techniques:

- **Time Diversity**: Multiple versions of the signal are transmitted at time spacings that exceed the coherence time of the channel. It is usually combined with forward error correction code. So multiple versions of coded signal are received with independent fading conditions. With the help of error correction codes, the transmission errors can be corrected.

- **Frequency Diversity**: Transmitting a signal on several carrier frequencies exposes the transmission to diversified effects of multipath propagation. If the frequencies are separated by more than the coherence bandwidth of the channel, they will have independent fading conditions. Typical implementations are spread spectrum and OFDM modulation with error correction coding across a large bandwidth.

- **Space Diversity**: The signal is transmitted over several propagation paths and arrives at multiple receiving antennas. If the antennas are spatially separated by several wavelengths (minimum $0.4\lambda$ where $\lambda$ is the wavelength corresponding to the carrier frequency [34]), each of the antennas has independent fading conditions. The advantage of space diversity is that it does not incur extra bandwidth or delay. Typical implementations are MISO, SIMO, MIMO.

- **Traffic Diversity**: Users usually do not make requests at the same time. Some users are transmitting the packets while others remain idle. If each user has its own independent data link, this results in an imbalanced traffic and underused bandwidth. However, if users pool
their links together and exploit the traffic diversity gain, it can improve the perceived throughput and network utilization rate.

### 1.2.3 Cooperation Approaches

To utilize these diversity gains, the proposed distributed cooperative framework consists of the following three basic strategies:

- **Signal Combining**: In this cooperation strategy, which is usually implemented at the physical/link layer (Figure 1.2), signals are received by all the nodes in cooperation, the assisting nodes forward the received signals to the master node, and the master node combines all the data to correct transmission errors. By exploiting the channel diversity gain and antenna gain, signal combining can significantly improve network robustness and capacity. A limitation of this approach is that it requires the modification of the physical layer and device hardware.

- **Traffic Multiplexing**: In this cooperation strategy, the cooperating nodes bundle their long-range low-rate links together to form a high speed link with aggregated capacity. The traffic from each node is spread among the links of all cooperating nodes (in other words, multiplexed in this bundled link). The advantage of adopting this strategy is that it provides load-balancing, and therefore achieves a higher network utilization, improves the robustness and perceived throughput through both traffic diversity and channel diversity. This approach can be implemented as a software middleware in the network layer, and has no impact on the physical layer and device hardware.
• **Cross-Layer Cooperation:** Signal combining and traffic multiplexing are orthogonal strategies, so they can be adopted independently. It is also possible to incorporate both to allow interactions between them. In the cross-layer cooperation, traffic scheduling can be designed on the basis of the channel condition. The traffic scheduler decides who should be included in the cooperation and how the traffic is being routed. For example, if one node sends a packet corrupted during the transmission, the re-transmission packet can be rerouted through another cooperating node with best channel conditions and possibly with a different coding scheme. Because of the resulting complexity of such coupling, and as a first step, our immediate research focuses on the previous two basic cooperation strategies. In the future, we will investigate the general cross-layer cooperation.

![Cooperation at multiple network layers.](image)

**Figure 1.2:** Cooperation at multiple network layers.
1.2.4 Modes of Operation

The proposed framework can operate in two modes: Master-Slave mode and the generalized Peer-to-Peer mode. In this thesis research, we will focus on investigating the Master-Slave mode. As a first step, this would help us understand the characteristics of each cooperation technique. The long term goal of this research is to develop practical solutions that can work in the Peer-to-Peer mode.

- **Master-Slave:** A node currently communicating over its long-range communication interface becomes a master. Surrounding nodes that are willing to cooperate, become assisting nodes or slaves. To achieve channel diversity, the master node collects contribution from the assisting nodes and dictates the cooperation strategy to satisfy the short-range communication requirements while minimizing the bit error rate of the combined signals at the master node.

- **Peer-to-Peer:** This cooperation happens according to a distributed multi-round and multi-hop algorithm. All the nodes are assumed to operate with the common goal to help their neighbours obtain higher throughput meanwhile being helped by others. The nodes structure themselves to achieve the best performance depending on the current topology, traffic and channel conditions, and available energy.

1.2.5 System Constraints

In our proposed diversity cooperative framework, a group of nearby nodes help each other by relaying the data through a local high-speed wireless
network using a separate air interface. Even though this local wireless network is much faster than the long-range cellular links, the total amount of available bandwidth is still limited.

The existing techniques introduced in the past for countering fading (e.g., Maximum Ratio Combining, and Generalized Selective Combining [21, 60, 34]) were designed for antennas that are wired to a central combiner and not restricted by the local communication limitations.

The proposed cooperation framework makes the nodes cooperate by relaying information to other nodes through a local wireless network. This raises questions on how to maximize the system performance while meeting the constraints of the short-range network bandwidth, computation and energy consumption. Given that the local bandwidth is shared with other nodes and local communications can also consume a significant amount of energy, it is critical to minimize the usage of the local bandwidth. This research aims at developing several cooperation techniques that improve the long-range communication performance while accounting for the local bandwidth constraints.
1.2.6 Security and Fairness

For the proposed cooperation to become popular, there is a need to develop mechanisms that reward cooperative nodes, detect and punish selfish behavior, and protect privacy [22, 30]. Such mechanisms are very important for the success of a distributed diversity system, but they are out of scope of this thesis research, and will be the subject of our future research. Our main focus is on understanding and demonstrating the potential of distributed diversity.

1.3 Evaluation and Experimentation

1.3.1 Metrics

In our analysis, we consider multiple evaluation metrics, which are enumerated below. Signal combining techniques such as the proposed Threshold Maximum Ratio Combining and Priority Maximum Ratio Combining aim at providing reliable communications, lower the outage probability and bit error rate, and improve the throughput. For traffic multiplexing techniques such as DLB, their goals are to improve network/spectrum efficiency and perceived throughput.

- **Outage Probability**: Outage probability is a commonly used metric to evaluate the performance of communication systems. In wireless fading channels such as the Rayleigh or Rician channels, the channel characteristics change rapidly. For acceptable communication performance, the signal received at the receiver needs to be above a min-
imum SNR. Failing to reach this threshold can prevent the receiver from decoding the signal, and thereby cause the outage. Let $\gamma_0$ be the minimum SNR that can be tolerated by the decoding scheme. $P_{\text{out}}$ is defined as

$$P_{\text{out}} = p(\gamma < \gamma_0) = \int_0^{\gamma_0} p(\gamma) \, d\gamma,$$

where $p(\gamma)$ is the probability distribution of SNR at the receiver.

- **Bit Error Rate (BER):** After the demodulation at the receiver, BER is the number of incorrect bits divided by the total number of bits transferred during a certain amount of time. For a single bit, BER means the probability of an error. BER is a basic metric of the performance of a wireless communication systems and it can be impacted by various factors: noise, interference, multipath fading, modulation, coding schemes, etc. In our research, to avoid the bias caused by all the possible channel coding schemes, we choose to focus on an uncoded link. We also choose to use the coherent Minimum-Shift Keying (MSK) as our modulation scheme in our analysis, due to its simple analytical model.

- **Throughput:** For a communication system, throughput is the average rate of successful packet/frame delivery. For a simple idealized protocol, a packet is a sequence of bits with a certain length. In this case, the packet error rate can be directly derived from the BER as $1 - (1 - p_b)^L$, where $p_b$ is the BER and $L$ is the packet length. In real world protocols, however, the frame has a specific format. For example, due to the nature of wireless communications packets it usually contains an error detection code and forward error correction (FEC) code to counter the transmission errors. So the throughput calculation also needs to take
the packet overhead into account. If we assume that the packet length is a constant, then the throughput can be represented as

$$ p_t = \frac{(L - OH)(1 - p_b)^L}{L}, $$

where $OH$ denotes the overhead length. If the packet length is not a constant, then the throughput is as following

$$ p_t = \sum_{l} \frac{(l - OH)(l - p_b)^l}{l} \cdot p_L(l), $$

where $L_{max}$ is the maximum possible packet length and $p_L(l)$ is the probability distribution of the packet with length $l$. Variable packet length is usually caused by the payload. For our research purpose, we simply consider the packets with a constant length, but we also believe that the results can be easily generalized to a variable packet length.

- **Average Service Delay**: This is the average amount of time that a packet stays in the system. The service delay is the waiting period plus the transmission period. The waiting period starts when a packet is dispatched from the application layer for transmission. If the queue is not empty or if the channel is occupied, the packet is put on hold until it reaches the head of the queue and the channel is cleared. The transmission period is the actual amount of time it takes to finish transferring the packet. This period depends on the packet length and the transmission rate.

- **Perceived Throughput**: In a DLB cooperation system with multiple nodes in cooperation, the perceived throughput is the expected ratio of request length and the processing time plus the waiting period. It can be defined as

$$ PT = E\left(\frac{L}{\mu' + D}\right). $$
where $L$ denotes the request length, $D$ is the waiting period before a request's processing starts, and $\mu'$ is the aggregate service rate.

1.3.2 Testbeds and Developed Prototypes

We have developed testbeds and prototypes to validate our assumptions and analysis results, and also serve as proofs of concept. We created a simulation testbed using Matlab and Simulink to simulate the proposed Priority Maximum Ratio Combining technique under fading environment (Section 4.6), and Distributed Link-Bonding under serial and parallel-multiplexing (Section 6.4). For the proposed Post Soft-Demodulation Combining technique, we create a real-world testing environment using GNU Radio and USRP. Although there are some limitations with the hardware, our experimental results still reveal a promising gain (Section 5.5.2).

We also created a couple of prototypes of the proposed DLB system. Our initial attempt is a tunnelling based proxy DLB system on laptop computers. It demonstrates the potential of the DLB system. However, it also raises some performance issues, which will be described in Section 7.2. For the second prototype – the Neptune DLB system, we solve the problems faced in the first prototype, and make it to run on the modern smart phones (Section 7.3).

1.4 Contributions

We explore the potential of distributed diversity and cooperation to improve the performance of the hybrid wireless networks. A hierarchical cooperative framework at multiple levels of the network stack and across the local ad
hoc network and the cellular network is proposed. A two-phase protocol that implement the hierarchical strategy is also developed.

Within the framework, we propose several distributed signal combining techniques: Threshold Maximum Ratio Combining [42], Priority Maximum Ratio Combining [43] and Post Soft-Demodulation Combining. Our analytical and experimental results show the proposed cooperative techniques are able to improve bit error rate and throughput by at least an order of magnitude in some typical cases with awareness of local resource constraints [56].

We propose a Distributed Link-Bonding system that exploits both channel diversity and traffic diversity. We show that the proposed strategy is a simple and effective method to improve the signal coverage, delays, perceived throughput and network utilization through analysis, simulations and experiments [44].

We also develop prototypes for the proposed Distributed Link-Bonding system. We show that the proposed methods can be implemented on current mobile phones, transparent to applications, and easy to use with simple user controls.

1.5 Related Work

In this section, we provide a quick overview of some of the research that has been done and that overlaps with the concepts, problems, and solutions that we address in this thesis. We discuss more in detail the work related to each specific aspect of our research in the corresponding chapters.

The fundamental diversity principle in wireless communications has been
studied in [62]. A variety of fading channel models are analysed in [34, 68]. Recently distributed cooperation for wireless communication has attracted significant interest from the research community [31, 61, 40, 15]. Some studies have addressed specific cases such as diversity with homogeneous interfaces where the combining occurs over the air [64, 50, 38]. A theory of distributed MIMO in ad hoc network has been studied in [58]. Cooperation of multi-radio access networks is advocated in [27] to enhance the transmission robustness. Smart antennas use an antenna array to optimize the preferred signal direction and increase system capacity [71]. Cognitive radios allow for the dynamic setting of radio parameters such as modulation and coding schemes [39]. Femtocells are introduced by several operators to extend service coverage indoors via broadband [72]. GNU Radio is an open-source software defined radio (SDR) platform, which enables signal processing in software. GNU Radio and USRP have become popular tools for prototyping wireless systems [5, 19].

Decoding with soft-decision values has been introduced in various error correction codes such as convolutional codes, turbo codes and LDPC to reduce the bit error rate [52, 33, 35]. Hybrid ARQ and packet combining are also effective ways to overcome communication errors by utilizing the corrupted packets [23, 29]. Space-time codes leverage both spatial diversity and temporal diversity by transmitting and receiving multiple copies of information across several antennas [14, 66].

Link Aggregation or Multi-Link Trunking (MLT) on the link layer for wired Ethernet is proposed in [7]. It allows to group multiple physical links into one logical Ethernet link to provide fault-tolerance and high-speed communication between routers. Multi-homed transport protocols such as the IETF Stream Control Transmission Protocol (SCTP) [65] provides the link redun-
1.5. RELATED WORK

dancy and multi-sessions. Indirect TCP or I-TCP [16] for mobile hosts splits the single TCP connection into two independent TCP links to prevent the performance fluctuation caused by wireless communications from propagating to the fixed network. Wireless Mesh Networks have also been proposed as a cooperative approach to wireless communications, and studied in [8, 11, 24]. It allows neighbors to connect their home networks together to share their Internet access via gateways that are distributed in their neighborhood. A multi-radio unification protocol for IEEE 802.11 wireless networks is proposed in [13]. Tor, an implementation of the second-generation Onion router, is a cooperative network system, which is used to protect users’ privacy and security while communicating on the Internet [28].
In this chapter, we introduce a distributed cross-layer channel diversity strategy - Hierarchical Priority Combining (HPC). We present the intuition and motivation for this type of cooperation in hybrid networks with long-range low data-rate links and short-range high data-rate links. We present the basic principle and mechanisms behind this strategy, its cooperation structure, and a two-phase protocol design. In the following chapters (Chapter 3–5), we provide an in-depth description of each of the combining techniques that comprise HPC. We analyse and compare their performance, and present the experimental results. The metrics we use to evaluate the proposed combining strategies include outage probability, bit error rate, throughput and local bandwidth usage.
2.1 Motivation and Technical Background

Cellular systems are designed for long-range and relatively low data-rate communications. Due to the long link distance, absence of line-of-sight (LOS), and mobility, they suffer from strong multipath fading and shadowing [62]. Recently, due to the increasing demand for mobile services such as mobile cloud computing and video streaming, improving the robustness and throughput of cellular systems has become critical. Simple solutions for reducing the bit error rate by increasing the transmission power would cause more interference and result in adversely affecting the system capacity. An alternative solution is to bring the cellular base stations closer to the mobile clients, but this requires a deployment of more base stations which is not economical and difficult to scale.

Diversity is an efficient and powerful tool to mitigate channel fading and shadowing [34, 68]. Most of the previous work on pre-demodulation diversity focuses on combining techniques using multiple antennas that are co-located on a single device. Given that all the antennas are directly interconnected, the combining algorithms have full access to the signals from all the antennas. However, co-locating multiple antennas on a single mobile device is impractical for the modern cellular systems. This is not only due to its high cost, but also because of the spatial separation needed between the antennas to achieve channel independence.

Recently, distributed cooperation has attracted more interest from the wireless communications and networking research community [31]. Some studies have addressed specific cases such as diversity with homogeneous interfaces where the combining occurs over the air [64, 50, 51, 49, 37, 38, 12]. Several interesting approaches demonstrated the benefits of distributed co-
operation in ad hoc networks with homogeneous wireless interfaces and challenged the community to investigate the full benefits of distributed cooperation [47, 46, 61, 40, 67, 31]. Theory of distributed MIMO in ad hoc network has also been studied in [58, 57]. The use of cooperation with heterogeneous air-interfaces was advocated in [15, 54]. Multi-radio access networks over link layer has also been researched in [27] to enhance the transmission robustness. More recently, several post soft-demodulation techniques were proposed for homogeneous air-interface systems [45, 41]. In our research, we consider the distributed cooperation with heterogeneous air-interfaces.

### 2.1.1 Motivation for Distributed Diversity

Nowadays, most smart-phones are equipped with a short-range high-speed local WiFi interface besides their long-range low data-rate cellular interface. Here, the high speed local network makes the distributed cooperation with a small group of nearby users possible. On one hand, the distance of cooperating users is close and usually within one hop of short-range radio communications, and the local short-range links are much faster and more reliable than the long-range cellular links. Those allow the neighbouring nodes to relay their received signals for each other through the local high-speed wireless network. Signals are eventually combined at the destination node to correct errors. On the other hand, users are also well separated, often meters away, which allows an independent transmission path from the cellular tower to each of the users. As a result, it is possible to implement the distributed cooperation to exploit both the antenna gain and the diversity gain. Furthermore, because wireless channel fading for different transmission paths are fairly independent, intuitively, only a few cooperative nodes
would be sufficient to correct transmission errors caused by fading. This is confirmed by our analysis results. However, in such a cooperative system, because the received data needs to be transmitted through the local wireless network, its bandwidth becomes a constraint. First, the cooperation traffic needs to fit within the available local bandwidth. Second, local communications can also require a significant amount of energy and computation, which are critical to the battery powered devices. Minimizing the amount of local communication and computation as well as maximizing the cooperation gain are the underlying design principles of our cooperation strategy.

In the rest of this section, we will present and discuss the system model and key mechanisms. In the next section, we will introduce a distributed multi-level signal combining strategy called Hierarchical Priority Combining (HPC), and present the underlying HPC protocol.

### 2.1.2 System Model

In our cooperative environment, there is one base station sending signal $s(t)$ to $M$ receiving nodes. Each of those receiving nodes are experiencing multipath fading. The receiving nodes cooperate with each other by relaying the signals through a local ad hoc wireless network using a separate radio interface. For a cooperation scheme based on pre-processed signals, the RF signal is down-converted, sampled and relayed (through the short-range interface) to the destination node for combining. In this case, we assume a coherent signal detection of each node for the phase shift, so we can combine the received signals from independent fading paths. For the cooperation schemes based on post-processed signals, the received signal at each node is first decoded independently and relayed to the destination node. In this
case, co-phasing is not required, and combining is done at the bit level.

In our analysis, we consider Rayleigh fading as our channel propagation model for cellular communications [34, 68]. As discussed in the previous chapter, Rayleigh fading is the typical model used for urban environments when no LOS is available and mobility is involved.

We now define it formally. Let $u(t)$ be the equivalent baseband signal

$$u(t) = u_I(t) + j u_Q(t),$$

where $u_I(t)$ and $u_Q(t)$ are the in-phase and quadrature components of signal $u(t)$, and $j$ is the imaginary unit.

So the transmitted signal $s(t)$ can be represented as

$$s(t) = Re\{u(t)e^{j2\pi f_c t}\} = u_I(t) \cos(2\pi f_c t) - u_Q(t) \sin(2\pi f_c t),$$

where $f_c$ is the carrier frequency.

On the receiver side, the received signal at the $i^{th}$ node can be represented as $r_i e^{j\theta_i} s(t)$, where $r_i$ is the signal amplitude and $\theta_i$ is the signal phase at node $i$. Assuming that the delay spread of each path is small relative to the inverse signal bandwidth of the transmitted signal, the received signal is the sum of all multipath components.

$$r(t) = Re\{u(t)e^{j2\pi f_c t}\} \left(\sum_k \alpha_k(t)e^{-j\phi_k(t)}\right)$$

$$= r_I(t) \cos(2\pi f_c t) - r_Q(t) \sin(2\pi f_c t)$$

$$r_I(t) = \sum_k \alpha_k(t) \cos \phi_k(t)$$

$$r_Q(t) = \sum_k \alpha_k(t) \sin \phi_k(t),$$
where $r_I(t)$ and $r_Q(t)$ are the in-phase and quadrature components of the received signal, and $\alpha_k$ and $\phi_k$ are the amplitude and the phase of the $k^{th}$ multipath component respectively.

Assume that $r_I$ and $r_Q$ are both zero-mean and equal variance Gaussian random variables, and $\phi_k$ is uniformly distributed on $[-\pi, \pi]$. It can be shown that the received signal envelope $|r(t)|$ is Rayleigh distributed and $r^2(t)$ is exponentially distributed [34].

$$|r(t)| = \sqrt{r_I^2(t) + r_Q^2(t)}$$

Let $N_0/2$ be the noise power spectral density (PSD). The SNR (signal to noise ratio) $\gamma = r^2(t)/N_0$ is exponentially distributed. Therefore, the probability distribution of the SNR is

$$p(\gamma) = \begin{cases} \frac{1}{\bar{\gamma}} e^{-\frac{\gamma}{\bar{\gamma}}}, & \gamma \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

where $\bar{\gamma}$ denotes the long run average SNR. The cumulative distribution function is

$$p(\gamma \leq t) = \int_0^t \frac{1}{\bar{\gamma}} e^{-\frac{\gamma}{\bar{\gamma}}} d\gamma$$

$$= 1 - e^{-\frac{t}{\bar{\gamma}}}$$

Since the cooperative nodes in this distributed environment are spatially separated by the distance of meters or tens of meters, which are many times longer than the carrier wavelength (for $f_c = 1800$MHz, the wavelength $\lambda = 0.16m$), it is reasonable to assume the independent fading path for each node. Under the same assumption of Rayleigh channel, we also assume that the fading for each propagation path is independent and identically
2.1. MOTIVATION AND TECHNICAL BACKGROUND

distributed (i.i.d.). Therefore, the probability of the signals received by all nodes having an SNR less than \( t \) is

\[
p(\gamma_1 \leq t, \ldots, \gamma_M \leq t) = p(\gamma_1 \leq t) \cdots p(\gamma_M \leq t) = (1 - e^{-\frac{t}{\bar{\gamma}}})^M
\]

This means that using multiple receivers can significantly reduce the probability of experiencing a strong fading. In the following, we show that the cooperation can result in even better performance when the signals from multiple receivers are combined.

2.1.3 Maximum Ratio Combining

Maximum Ratio Combining (MRC) is an effective technique to combine multiple independent signal sources so that the effects of multipath fading are mitigated. MRC is a linear combining technique. The output of the MRC combiner is a weighted sum of all branches [21].

Let \( n \) be the total number of signal sources or branches for combining. The signal received from the \( i^{th} \) source is \( r_i e^{j\theta_i} s(t) \), where \( r_i \) is signal amplitude and \( \theta_i \) is the signal phase. MRC makes all the signals from the considered branches co-phased. Thus, a weight \( \alpha_i = a_i e^{-j\theta_i} s(t) \) is applied at the \( i^{th} \) signal source. Therefore, we have the MRC output as

\[
r = a_1 r_1 + \cdots + a_n r_n
\]

Let’s assume that the fading channels are i.i.d. with equal noise power spectral density \( N_0/2 \). Then, the SNR of the combined signal is

\[
\gamma_{\Sigma} = \frac{(\sum_i a_i r_i)^2}{N_0 \sum_i a_i^2}
\]
Let $\gamma_i$ be the random variable for the SNR from the $i^{th}$ source. It can be shown that the combined signal SNR $\gamma_{\Sigma}$ is maximized if the $a_i$ weights are chosen to be the square root of the SNR from each source (i.e., $a_i = \sqrt{\frac{r_i^2}{N_0}}$) [34], which leads to the combined SNR

$$\gamma_{\Sigma} = \gamma_1 + \cdots + \gamma_n$$

Therefore, the SNR performance of MRC scales linearly with the number of independently contributing signal sources.

Although the MRC technique is effective, it needs to be implemented on a single device with a directly attached antenna array. If it were the wired case, it would not be a problem due to the virtually unlimited local bandwidth. However, this is not the case of our distributed environment. Hence, the questions we try to investigate are: What are the possible coop-
2.2. HIERARCHICAL PRIORITY COMBINING

Operation techniques that can be used to mitigate the channel fading effects while accounting for the limited local bandwidth and energy? What are the trade-offs between performance improvement and local bandwidth usage and energy constraints? What network protocols and architectures do we need to support this cooperation? In the next section, we will answer those questions by introducing the Hierarchical Priority Combining strategy and several techniques used within the strategy.

2.2 Hierarchical Priority Combining

Hierarchical Priority Combining (HPC) incorporates three levels of combining techniques that have different error correction capabilities and local bandwidth requirements. We first outline the three basic combining techniques, and then describe the proposed Hierarchical Priority Combining protocol.

2.2.1 The Hierarchical Structure

- **Decode-and-Forward**: If at least one of the assisting nodes can demodulate the packet and verify its integrity, then the decoded packet can be relayed to the master node through its short-range link. This level of combining uses the minimum local bandwidth, but can only be used when the overall signal strength is high, and the mobile nodes are experiencing strong uneven fading or shadowing. This could be the case when a group of people are in motion, e.g., inside a car, a bus, or a train. A similar idea has been discussed in [51]. The main difference in our research is that we are considering to relay the packet through
a different interface rather than re-injecting it back to the same channel with a different coding scheme. This approach, in our opinion, is more realistic from system’s perspective, however it requires a different analysis.

- **Post Soft-Demodulation Combining:** At this level, the signal received by each of the assisting nodes has incorrectable errors. However, it is already strong enough for demodulation. In this case, *some* of the assisting nodes with the *strongest* received signals, send the soft-decision output of the demodulator to the master node for bit-level combining (Refer Chapter 5 for more detail of soft-decision values). Cooperation at this level can be very efficient at correcting errors when the signal strength is relatively high. This is still a sub-optimal diversity combining technique but has the advantage of requiring only a moderate short-range communication bandwidth.

- **Pre-Demodulation Combining:** At this level, *some* of the assisting nodes transmit the sampled down-converted RF-signal to the master node. We introduce *Threshold Maximum Ratio Combining* (TMRC) and *Priority Maximum Ratio Combining* (PMRC) as the potential candidates for Pre-Demodulation Combining. For TMRC, the assisting nodes with SNR above a pre-set threshold relay their received signals to the master. For PMRC, only the assisting nodes with the strongest SNR relay their received signals to the master. The master then combines its received signal with other gathered signals. Signal combining at this level gives the best error correction capability, but communicating the digitized waveform information requires a large local bandwidth. Therefore, it is more appropriate for the scenarios where the long-
range radio signal is extremely weak and experiences strong fading, but the local short-range links is fast and stable.

### 2.2.2 Two-Phase Protocol Design

The HPC protocol dynamically decides which of the above three combining techniques to use at the time of the reception of the packet. Intuitively, Pre-Demodulation Combining takes all the information of the originally received signals among the cooperative nodes, so it should perform the best in error correction, but it also requires a huge amount of local bandwidth. Decode-and-Forward Combining and Post Soft-Demodulation Combining can be taken as the light-weight version of Pre-Demodulation Combining as it either sends the complete demodulated data or partially demodulated data with soft-decision values. The advantage of Decode-and-Forward Combining is that it uses a minimal amount of local bandwidth, but requires a node to have a strong signal reception in order to independently and successfully decode the packet. Post Soft-Demodulation Combining, on the other hand, performs better due to the freedom of using soft-decision values from multiple sources compared to Decode-and-Forward Combining.

To achieve the best performance while still minimizing the local bandwidth usage, our HPC strategy uses the received signal quality to decide which combining technique to adopt for each packet. There are many possible ways to cooperate, but the proposed HPC strategy has the benefit of being effective and easy to implement due to its simplicity and hierarchical structure.

The HPC cooperation protocol runs in two phases. *Phase I* is a very short period, within which the nodes exchange information with each other about
the quality of received signal. In Phase II, each node decides if and what level of combining information it will send to the master. In the following, we provide a high-level description of the protocol.

Let $M$ be the total number of nodes involved in the cooperation, and $N$ be the number of signal sources involved in combining. Note that since the cooperation always includes the master node, $N - 1$ is the actual number of distributed assisting nodes that relay their signals to the master node.

**Phase I:** The master node broadcasts a cooperation request beacon if it is unable to decode the packet. Upon receipt of the cooperation-request beacon, the assisting nodes measure the SNR of the received signal (denoted by $\gamma$) from their long-range air interface and compare it with a predefined threshold $\gamma_D$ ($\gamma_D$ is the threshold above which demodulating the packet is feasible.) If $\gamma < \gamma_D$, the assisting nodes broadcast the SNR to others. Otherwise, they will try to demodulate the packet independently and verify its integrity using a CRC-like checksum. Finally, the assisting nodes broadcast both the SNR and the CRC verification result. Each node is assigned a particular time slot during the phase I to avoid collision.

**Phase II:** In this phase each node makes a decision after hearing the report of signal quality from other assisting nodes. If at least one assisting node can demodulate the long-range RF signal and pass the CRC check, one of them with the highest ID will relay the decoded packet to the master, that is the Decode-and-Forward case. If no one passes the CRC check and the total number of assisting nodes with $\gamma > \gamma_D$ is more than a predefined value, the top $N_{\text{soft}} - 1$ nodes with the strongest SNR transmit (in the order of their ID) their soft-decision values to the master for Post Soft-Demodulation Combining. $N_{\text{soft}}$ is a pre-set system parameter, and the transmission size
2.2. HIERARCHICAL PRIORITY COMBINING

$N_{\text{soft}} - 1$ is limited by the local bandwidth. In the end, if none of the above cases happens, then the assisting nodes, according to TMRC or PMRC protocol, send the sampled long-range radio waveform to the master node for Pre-Demodulation Combining.

In the following chapters, we will present several combining techniques at different level of HPC and analyse their performance separately. As we will show, for each combining technique, $N$ can be much smaller than $M$, and it still achieves most of the combining gain. To derive the overall performance of HPC as a global hierarchical combining strategy can be difficult, due to the complexity introduced by system parameters such as $\gamma_D$ and $N_{\text{soft}}$. There is no straight way to model them all together, so we suggest building a system that implements HPC and tune those parameters in the real world. This will be the part of our future work.
CHAPTER 3

Threshold Maximum Ratio Combining

In this chapter, we introduce a pre-demodulation signal-combining technique called Threshold Maximum Ratio Combining (TMRC). In the MRC scenario, the branches with high SNR get the highest weights and the branches with low SNR get the lowest weights. If the local resources are limited, a natural modification of MRC would be to keep only the high SNR branches and discard the lower ones. TMRC is an extension of MRC, adopting the idea that accounts for the limited local bandwidth available in distributed cooperation. In this chapter, we analyse TMRC performance and discuss its limitations.

3.1 Introduction

We consider a system of one base station and $M$ cooperative mobile stations including a master node. In TMRC, $M - 1$ assisting nodes relay their sampled signals to the master node if and only if their SNR is above a threshold $\gamma_T$, which is a system parameter and set according to the channel condition and available local bandwidth. The master node then collects the signals from
all the assisting nodes and combines them using Maximum Ratio Combining (MRC) [60, 34, 68].

If $\gamma_T$ is set to zero, this implies all the nodes are relaying their received signals to the master all the time, which is the case of the traditional $M$-MRC (MRC with $M$ branches). If $\gamma_T$ is set to $+\infty$, none of the assisting nodes is relaying its received signal and the master can only use the signal from itself, which is the case of no cooperation. From the analysis, we show that with proper selection of the threshold $\gamma_T$, TMRC can provide the benefit of diversity gain like MRC but still limit the local communication.

In order to study the performance of TMRC, we first analyse the distribution of the SNR of the combined signal. For $M$ nodes in cooperation with the threshold $\gamma_T$, we will show:

$$p_{\text{TMRC}}(\gamma) = \sum_{i=1}^{M} \left( \frac{M-1}{i-1} \right) \cdot C^{M-i} \cdot g^{(i)}(\gamma), \quad \gamma \geq 0$$

where

$$g^{(i)}(x) = \frac{(x - (i-1) \cdot \gamma_T)^{i-1}}{(i-1)! \cdot \gamma_T^i} \cdot e^{-\frac{x}{\gamma_T}},$$

for $x \geq (i-1) \cdot \gamma_T$ and $C = 1 - e^{-\frac{\gamma_T}{\gamma_T}}$ (See Theorem 3.2.5).

Based on this distribution, we can further derive the performance of TMRC, e.g., outage probability, bit error rate, throughput, bandwidth and energy consumption.

The structure of the proof is as follows. We first derive the signal distribution from each assisting node. Theorem 3.2.3 shows the probability distribution of the combined SNR for a simplified case in which all the nodes (including the master node) only use the signal above the threshold $\gamma_T$. Theorem 3.2.5 gives the distribution of the combined SNR for a practical proto-
col, which differentiates the master node from the assisting nodes in the way that the master node uses the signal in the full SNR range, as no consumption of the local bandwidth incurs. And $g^{(i)}(x)$ is derived in Lemma 3.2.4.

### 3.2 Probability Distribution of SNR for TMRC

To derive the probability distribution of SNR, we consider the typical multipath fading model - Rayleigh fading, and we keep the MRC assumption that the nodes long-range signals’ SNRs are i.i.d. with parameter $\bar{\gamma}$ and $p(\gamma) = \frac{1}{\bar{\gamma}} e^{-\frac{\gamma}{\bar{\gamma}}}$, where $\gamma \geq 0$. In TMRC, each assisting node relays the signal if and only if its signal SNR above the threshold $\gamma_T$. The master node collects the signals from each assisting node. Let $p_T(\gamma)$ be the distribution for each branch at the master side. We have $p_T(\gamma)$ equals $p(\gamma)$ when $\gamma \geq \gamma_T$, and $p_T(\gamma)$ equals 0 when $0 < \gamma < \gamma_T$ because the master will not get any signal from the assisting node in that range. The formal definition of $p_T(\gamma)$ is given in Equation 3.2.

![Figure 3.1: PDF of $\gamma$ of each branch at master side in TMRC](image)

Let $\delta(x)$ denote the Dirac delta function defined as follows

$$\delta(x) = \begin{cases} \infty, & x = 0 \\ 0, & x \neq 0 \end{cases}$$

and $\int_{-\infty}^{+\infty} \delta(x) \, dx = 1$. 

CHAPTER 3. THRESHOLD MAXIMUM RATIO COMBINING

The distribution of the SNR $T$ for each branch under TMRC, as illustrated in Figure 3.1, therefore is

$$p_T(\gamma) = \begin{cases} 
(\int_0^{\gamma_T} \frac{1}{\gamma} e^{-\frac{x}{\gamma}} \, dx) \cdot \delta(\gamma), & \gamma = 0 \\
0, & 0 < \gamma < \gamma_T \\
\frac{1}{\gamma} e^{-\frac{\gamma}{\gamma_T}}, & \gamma \geq \gamma_T 
\end{cases}$$

Let $f(x) = \begin{cases} 
\frac{1}{\gamma} e^{-\frac{x}{\gamma}}, & x \geq \gamma_T \\
0, & x < \gamma_T 
\end{cases}$ and $C = \int_0^{\gamma_T} \frac{1}{\gamma} e^{-\frac{x}{\gamma}} \, dx = 1 - e^{-\frac{\gamma}{\gamma_T}}$.

Thus, the probability distribution of the SNR for each branch can be written as

$$p_T(\gamma) = C \cdot \delta(\gamma) + f(\gamma) \quad (3.2)$$

**Definition 3.2.1.** For a function $h(x)$, recursively define $h^{(i)}(x)$, where $i \geq 0$, as following, $h^{(0)}(x) = 1$, $h^{(1)}(x) = h(x)$ and $h^{(i)}(x) = h(x) \ast h^{(i-1)}(x)$, where $\ast$ is the convolution operator.

**Lemma 3.2.2.**

$$f^{(i)}(x) = \frac{(x - i \cdot \gamma_T)^{i-1}}{(i-1)! \cdot \gamma_T^i} \cdot e^{-\frac{x}{\gamma_T}}, \quad \text{for } x \geq i \cdot \gamma_T, \quad i \geq 1 \quad (3.3)$$

**Proof.** By induction.

Base case: $i = 1$,

$$f^{(1)}(x) = \frac{(x - \gamma_T)^0}{0! \cdot \gamma_T} \cdot e^{-\frac{x}{\gamma_T}} = \frac{1}{\gamma_T} e^{-\frac{x}{\gamma_T}}, \quad x \geq \gamma_T$$

Hypothesis: assume Equation 3.3 holds for $i = j$.

For the case $i = j + 1$,
According to the definition of \( f^{(j+1)}(x) \), we have

\[
f^{(j+1)}(x) = f(x) * f^{(j)}(x)
\]

\[
= \int_{-\infty}^{\infty} f(x) \cdot f^{(j)}(x - \tau) \, d\tau, \quad \tau \geq \gamma_T; \quad x - \tau \geq j \cdot \gamma_T
\]

\[
= \int_{-\infty}^{\infty} \left( \frac{1}{\gamma} e^{-\frac{x}{\gamma}} \right) \cdot \left( \frac{(x - \tau - j \cdot \gamma_T)^{j-1}}{(j-1)! \cdot \gamma^j} \cdot e^{-\frac{x-\tau}{\gamma}} \right) \, d\tau
\]

\[
= \frac{(x - (j+1) \cdot \gamma_T)^j}{j! \cdot \gamma^{j+1}} \cdot e^{-\frac{x}{\gamma}}, \quad x \geq (j+1) \cdot \gamma_T
\]

Theorem 3.2.3. The distribution of \( \gamma_{\Sigma_M} \), which is the sum of \( M \) channels under i.i.d. Rayleigh fading with threshold \( \gamma_T \) is:

\[
p_{\Sigma_M}(\gamma) = p_{\Sigma}^{(M)}(\gamma) = (C \cdot \delta(\gamma) + f(\gamma))^{(M)}
\]

Proof. Expand the equation (3.4) and simplify using the Dirac function property:

\[
\delta(x) * f(x) = f(x)
\]

Let \( g^{(i)}(x) = \)

\[
\begin{cases}
p(x) * f^{(i-1)}(x), & x \geq (i-1) \cdot \gamma_T \\
0, & x < (i-1) \cdot \gamma_T , \quad i \geq 1
\end{cases}
\]
Lemma 3.2.4.

\[ g^{(i)}(x) = \frac{(x - (i - 1) \cdot \gamma_T)^{i-1}}{(i-1)! \cdot \gamma^i} \cdot e^{-\frac{x}{\gamma}}, \quad \text{for } x \geq (i - 1) \cdot \gamma_T, \quad i \geq 1 \] (3.6)

Proof.

\[
\begin{align*}
g^{(i)}(x) &= p(x) \ast f^{(i-1)}(x) \\
&= \int_0^{x-(i-1)\cdot\gamma_T} p(\tau) \cdot f^{(i-1)}(x-\tau) \, d\tau, \quad \tau \geq 0; x-\tau \geq (i-1) \cdot \gamma_T \\
&= \int_0^{x-(i-1)\cdot\gamma_T} \left(\frac{1}{\gamma} e^{-\frac{x}{\gamma}}\right) \cdot \frac{(x-\tau-(i-1)\cdot\gamma_T)^{i-2}}{(i-2)! \cdot \gamma^{i-1}} \cdot e^{-\frac{x-\tau}{\gamma}} \, d\tau \\
&= \frac{(x - (i - 1) \cdot \gamma_T)^{i-1}}{(i-1)! \cdot \gamma^i} \cdot e^{-\frac{x}{\gamma}}, \quad x \geq (i - 1) \cdot \gamma_T
\end{align*}
\]

To get the distribution \( p_{TMRC}(\gamma) \) of the cooperative network with size \( M \), we need to add the \( \gamma \) of the master node to the \( \gamma_{\Sigma M} \) of the \( M - 1 \) assisting nodes.

Theorem 3.2.5. \( \gamma_{TMRC} \) is the sum of \( M \) channels under i.i.d. Rayleigh fading in which \( M - 1 \) branches are with threshold \( \gamma_T \) and one master node which always uses its received signal, i.e., \( \gamma_{TMRC} = \gamma + \gamma_{\Sigma M - 1} \). The distribution of \( \gamma_{TMRC} \) is:

\[
p_{TMRC}(\gamma) = \sum_{i=1}^{M} \binom{M - 1}{i - 1} \cdot C^{M - i} \cdot g^{(i)}(\gamma), \quad \gamma \geq 0 \] (3.7)
3.3. PERFORMANCE ANALYSIS

Proof:

\[
T_{RMRC}(\gamma) = p(\gamma) \ast T_{\Sigma M-1}(\gamma) \\
= p(\gamma) \ast \left( \sum_{i=0}^{M-1} \binom{M-1}{i} \cdot C^{M-1-i} \cdot f^{(i)}(\gamma) \right) \\
= \sum_{i=1}^{M} \binom{M-1}{i-1} \cdot C^{M-i} \cdot g^{(i)}(\gamma), \quad \gamma \geq 0
\]

By taking the limit of \(\gamma_T \to 0\), we get

\[
\lim_{\gamma_T \to 0} T_{TMRC}(\gamma) = \frac{\gamma^{M-1}}{(M-1)!} \cdot \epsilon^{\frac{\gamma}{\bar{\gamma}}}.
\]

It is also the distribution of \(\gamma_{\Sigma}\) of the traditional MRC. Therefore, MRC can be viewed as a special case of TMRC with the threshold zero.

3.3 Performance Analysis

Our distributed cooperative system is limited by the available local bandwidth. For wireless communication, transmission accounts for most of the energy consumption. So, for this part of the analysis, we will measure the amount of local transmission and its impact on the performance. Since MRC achieves the full diversity order and TMRC as a variant of MRC intentionally discards the low SNR signals at each assisting nodes, intuitively its performance should be in-between MRC and no cooperation. TMRC has the advantage of being able to satisfy the local bandwidth requirements. We discuss the trade-off between the performance and bandwidth usage in terms of outage probability, bit error rate and throughput. All the plots in the following are based on the analysis in the preceding section.
3.3.1 Outage Probability

From Section 1.3.1, let $\gamma_0$ be the minimum SNR for acceptable performance of the demodulator. The outage probability is defined as

$$P_{\text{out}} = P_{\text{TMRC}}(\gamma < \gamma_0) = \int_{0}^{\gamma_0} p_{\text{TMRC}}(\gamma) \, d\gamma,$$

where $p_{\text{TMRC}}$ is derived in Theorem 3.2.5.

![Figure 3.2](image1.png)  ![Figure 3.3](image2.png)

**Figure 3.2:** The outage prob. of the master node in TMRC ($\gamma_T = \gamma_0$) with $M = 2, 3, 4, 5$ compared to non-cooperative case $M = 1$.

**Figure 3.3:** The outage prob. of the master node with different $\gamma_T$ in TMRC compared to MRC and non-cooperative $M = 1$.

When set $\gamma_T = 0$, the outage probability of the master node in TMRC is the same as the one in MRC. As we increase $\gamma_T$ the outage probability increases until $\gamma_T$ reaches infinity where it becomes the case of no cooperation. This is reasonable because when $\gamma_T$ is set to infinity no assisting node transmits; the master only uses its own received signal. Figure 3.2 shows the outage probability of the master node in TMRC with various number of nodes in cooperation. We set $\gamma_T = \gamma_0$ for all cases. As we can see, the outage probability decreases significantly as the number of nodes ($M$) increases. Figure 3.3 shows examples of TMRC ($M = 5$, i.e., 4 assisting nodes)
3.3. PERFORMANCE ANALYSIS

with different $\gamma_T$, MRC (full diversity $M = 5$) and the non-cooperative case ($M = 1$). Note that if $\gamma_T$ is in the order of $\gamma_0$, the outage probability of TMRC is very close to MRC. For a target outage probability $10^{-2}$, if $\gamma_T$ is set to $\gamma_0$, the transmitter only needs to transmit at the power $17dB$ lower than the non-cooperative case. For the case of $\gamma_T = 10\gamma_0$, the transmitter can still save as much as $9dB$ in transmission power.

3.3.2 Bit Error Rate

With the probability distribution of the SNR of TMRC derived in Theorem 3.2.5, we can further calculate the bit error rate of the TMRC system. Here we consider the coherent Minimum-Shift Keying (MSK) modulation, which is similar to GMSK used in GSM system, with uncoded communication. We also assume a pulse shaping transmission with bit duration equal to $1/W$ such as raised cosine pulses \(^1\) with $\beta = 1$ (where $W$ is the used frequency bandwidth). These are commonly used assumption for estimating the BER of communication systems [34]. Therefore $E_b/N_0 = \gamma$ and the Bit Error Rate can be calculated as

$$BER = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) = Q\left(\sqrt{2\gamma}\right)$$

\(^1\)Similar to sinc(), but it is widely used in practice [60].
a target BER of $10^{-3}$, TMRC with 5 nodes and the $\gamma_T = 5$ dB requires 16 dB less power than the non-cooperative case. Higher gains are achievable when the target BER is lower. Figure 3.5, shows the impact of increasing number of nodes in cooperation. Similar to the outage probability, increasing the number of cooperative nodes improves the performance significantly. Notice that the improvement of the BER will directly translate to the Frame Error Rate ($FER = 1 - (1 - BER)^L$, where $L$ is the packet size).

### 3.3.3 Throughput

We can compute the throughput described in Section 1.3.1. For simplicity, we only consider an overhead of CRC (32 bits in this case) for packet error detection. Because this overhead is amortized over the whole packet, the packet size has also an impact on the net throughput. For the fair comparison of the various TMRC schemes, we use the packet size that maximizes the normalized throughput. Figure 3.6 shows that the throughput of the mas-
ter node can be largely increased by using signal combining with a limited
number of cooperative nodes. In TMRC the throughput can be improved by
either increasing the number of cooperative nodes or by lowering the thresh-
old $\gamma_T$, but both would cause an increase in the local bandwidth usage. For
example, if $M = 3$ and $E_b/N_0 = 10dB$ and we lower the $\gamma_T$ from $10dB$ to
$5dB$, then the throughput can be raised from 0.3 to 0.6 (a 100% growth).
We can also improve the throughput by introducing more cooperative nodes.
For example, if we add 2 extra cooperative nodes $M = 5$, the throughput is
raised to 0.5 from 0.3 (a 66% growth).
CHAPTER 3. THRESHOLD MAXIMUM RATIO COMBINING

3.3.4 Local Bandwidth Usage

The increase of the average SNR of the combined signal over the average SNR of each branch is called Energy Gain or Array Gain. Unlike the traditional MRC, in our distributed environment the local bandwidth is limited. TMRC uses $\gamma_T$ as a parameter to limit the local communication. In an extreme case, if $\gamma_T$ is set to 0, all the assisting nodes would constantly relay their received signals to the master node, so TMRC becomes MRC. If $\gamma_T$ is set to infinity, no assisting node would relay the signals, and thus it consumes zero local bandwidth. In this case, TMRC provides no performance improvement.

![Figure 3.7: Local bandwidth consumption under different $E_b/N_0$ in TMRC ($M = 5$).](image1)

![Figure 3.8: Bit error rate vs local bandwidth usage in TMRC ($\gamma_T = 5dB$, M = 5).](image2)

Figure 3.7 demonstrates the local bandwidth consumption of a TMRC system, with 5 nodes in cooperation, under multiple $E_b/N_0$. The local bandwidth is represented as the average number of assisting nodes that are transmitting their received signals to the master. As we can see, if $\gamma_T$ increases, the local bandwidth usage drops sharply. We also observe that higher $E_b/N_0$...
3.4. Protocol Implementation of TMRC

The packet containing the sampled signal is large and requires significant bandwidth for transfer over the local interface. Even though in TMRC the low SNR signal is dropped, as part of the HPC strategy, we can further improve the local bandwidth efficiency by introducing another threshold $\gamma_D$. When an assisting node finds that the received signal SNR is beyond $\gamma_D$, consumes more local bandwidth. Figure 3.8 shows the relation between the local bandwidth usage and bit error rate. As the average SNR increases, the bit error rate drops, but the local bandwidth grows too. It also suggests that with the same $\gamma_T$ the drop percentage of energy gain is always less than that of bandwidth requirement. It means we can always lower the amount of wireless communication with less loss of energy gain. In Figure 3.9, for the case of $M = 5$, if we set $\gamma_T = \bar{\gamma}$ which is at position 0, then the energy gain of TMRC is 79% of MRC while it needs only 36% of the local bandwidth of MRC. This provides a good justification for the performance and bandwidth trade-off in TMRC.

![Figure 3.9: Percentages of the energy gain and the bandwidth requirement of TMRC over MRC ($M = 5$).](image)
it assumes that the signal is good enough for demodulation and does not need any further combining. So it sends the demodulated bits to the master meanwhile informing other assisting nodes to refrain from sending their signal. Once the master receives the demodulated data, it uses it without combining with other signal.

In the general case, the implementation of the above strategy can be difficult and complex. The major reasons are: the channel characteristics may not be known in real time, the computation capability and battery on each mobile node might be limited, and the local network requires a fast MAC protocol, etc. However, we can design a simple abstract protocol (See Algorithm 1 and Algorithm 2) if we assume that the channel coherence time is larger than the duration of the timeslot (as noted in [17]). To simplify the MAC operation, the protocol introduces a short bit mapping period which allows the assisting nodes to indicate if they will transmit. That period is designed to be very short such that it does not impact the analysis of our system. Figure 3.10 shows a snapshot of the running protocol on a TDMA communication system. For every timeslot, the assisting nodes indicate their signal quality during the bit mapping period, and relay the sampled signal or the decoded data depending on if the signal can be decoded correctly.

![Figure 3.10: A running instant of TMRC with 1 master node and 3 assisting nodes. (I) The long-range TDMA channel (II) The short-range channel.](image-url)
### Algorithm 1: TMRC - Master Node Protocol

**Initialize the local cooperative network**

broadcast the control packet \{CINFO, $\gamma_T$, $\gamma_D$ and $G$\}

/* CINFO specifies the frequency, modulation and time slot, etc. */

/* $G$ is a set of nodes which will be active in this session for helping the master node. */

```plaintext
buf;  /* a queue to save the received signals */
```

Start the following two threads

**Thread 1: for the long-range interface**

```plaintext
while until the session ends do
  data_{lr} ← receive signal at next expected time slot
  measure $\gamma$
  enqueue(buf, [data_{lr}, $\gamma$])
end
```

**Thread 2: for the local interface and combining**

```plaintext
while until the session ends and buf becomes empty do
  [data_{buf}, $\gamma$] ← dequeue(buf);  /* block if queue is empty */
  if $\gamma \geq \gamma_D$ then
    demodulate(data_{buf}) and pass it to upper layer
    continue
  /* bm is the bit mapping structure */
  bm ← receive the bit mapping for the current time slot
  if bm indicates a demodulation from an assisting node then
    receive the demodulated data and pass it to upper layer
    continue
  if bm indicates at least one over threshold receiving then
    data_{loc[ ]} ← receive signals from each node sequentially according to the bit mapping
    data_out ← mrc_combine(data_{buf}, data_{loc[ ]})
  else
    data_out ← data_{buf}
    demodulate(data_out) and pass it to the upper layer
end
Algorithm 2: TMRC - Assisting Node Protocol

Join the cooperative network
Receive the control packet \{CINFO, \(\gamma_T, \gamma_D\), and \(G\)}

if the current node is not in set \(G\) then
  go to inactive mode (It’s excluded from the current session)

\(\text{buf} ;\) /* a queue to save the received signals */

Start the following two threads

Thread 1: for the long-range interface

begin
  while until session ends do
    data\_lr ← receive signal at next expected time slot
    measure \(\gamma\)
    enqueue(buf, \{data\_lr, \(\gamma\)\})
  end

Thread 2: for the local interface

begin
  while until session ends and buf becomes empty do
    \{data\_buf, \(\gamma\)\} ← dequeue(buf); /* block if queue is empty */
    Wait until the next bit mapping slot time
    if \(\gamma \geq \gamma_D\) then indicate a demodulation in the bit mapping slot
    else if \(\gamma \geq \gamma_T\) then indicate a over threshold receiving
    else
      indicate nothing in the bit mapping slot and discard the packet
    continue
    waiting until the bit mapping slot ends
    if the current node is the first node indicating a demodulation then
demodulate(data\_buf) and send the demodulated data
    else if no others indicate a demodulation then
      Waiting a period to allow other nodes which are indicating an over
      threshold receiving before the current node to finish their transmissions
      Send the sampled signal
    else discard the packet
  end
3.5 Generalization of TMRC

TMRC can be generalized to a class of new techniques called Randomized Maximum Ratio Combining (RMRC). In RMRC each node transmits the sampled signal to the master in a randomized manner with the probability determined by the signal SNR.

Let $T_X(\gamma)$ denote the function generating the probability of transmission. It takes the SNR as the input and outputs the probability of replaying the signal to the master. First, let $T_X(\gamma)$ be an exponential function, Figure 3.11a, $T_X'(\gamma) = 1 - e^{-c' \gamma}$, where $c'$ is a parameter. When $\gamma$ is low, the probability of transmitting the sampled signal is small. As $\gamma$ increases, the probability of transmitting goes up by following an exponential function. Similarly, we can define $T_X(\gamma)$ to be a linear function, Figure 3.11b, $T_X''(\gamma) = c'' \cdot \gamma$ if $\gamma < 1/c''$; $T_X''(\gamma) = 1$ if $\gamma \geq 1/c''$, where $c''$ is a parameter. A careful look at TMRC, indicates that it is just a special case of RMRC. In this case, Figure 3.11c, $T_X'''(\gamma) = 0$ if $\gamma < \gamma_T$; $T_X'''(\gamma) = 1$ if $\gamma \geq \gamma_T$.

RMRC techniques can be analyzed in the same manner as TMRC. For example, we can get the distribution of $\gamma$ from each node for different RMRC as following.

For $T_X'(\gamma)$, we have

$$p(\gamma) = C' \cdot \delta(\gamma) + \frac{1}{\gamma} e^{-\frac{\gamma}{c'} \cdot (1 - e^{-c' \gamma})},$$

$$C' = \int_0^{+\infty} \frac{1}{\gamma} e^{-\frac{\gamma}{c'} \cdot e^{-c' \tau}} d\tau = \frac{1}{c' \cdot \gamma + 1}$$

For $T_X''(\gamma)$, we have

$$p(\gamma) = \begin{cases} C'' \cdot \delta(\gamma) + \frac{1}{\gamma} e^{-\frac{\gamma}{c''} \cdot (c'' \cdot \gamma)}, & \gamma < \frac{1}{c''} \\ \frac{1}{\gamma} e^{-\frac{\gamma}{c''}}, & \gamma \geq \frac{1}{c''} \end{cases}$$
CHAPTER 3. THRESHOLD MAXIMUM RATIO COMBINING

\[ C'' = \int_0^{\frac{1}{\bar{\gamma}}} \frac{1}{\bar{\gamma}} e^{-\frac{\bar{\gamma}}{\bar{\gamma}}(1 - c'' \cdot \tau)} d\tau = c''\bar{\gamma} \cdot e^{-\frac{1}{c'' \cdot \bar{\gamma}}} - c'' \cdot \bar{\gamma} + 1 \]

(a) \( TX(\gamma) = 1 - e^{-c'\gamma} \)

(b) \( TX(\gamma) = c''\gamma \) if \( \gamma < 1/c'' \); 1 else

(c) \( TX(\gamma) = 0 \) if \( \gamma < \gamma_T \); 1 else

Figure 3.11: Various \( TX(\gamma) \) functions for RMRC.

3.6 Summary on TMRC and RMRC

In this chapter, we proposed TMRC and RMRC as two simple generalizations of MRC. We analytically derived their performance in terms of outage probability, bit error rate, throughput, and local bandwidth usage. We also proposed a possible practical protocol implementation. We showed when the threshold \( \gamma_T \) or the \( TX \) function is properly set, TMRC and RMRC can provide significant improvements of the BER and outage probability, still at
a reasonable local communication cost. In the next chapter, we will discuss choosing the proper threshold or the $TX$ function can be difficult, and propose an alternative cooperation mechanism without the drawbacks of TMRC and RMRC.
CHAPTER 4

Priority Maximum Ratio Combining

In this chapter, we introduce another pre-demodulation signal-combining technique - Priority Maximum-Ratio Combining (PMRC). It overcomes some of the limitations of TMRC. We evaluate its performance by analytically deriving the outage probability, bit error rate and throughput, and comparing them with TMRC. In the end, we present a protocol implementation and simulation results.

4.1 Introduction

TMRC, proposed in Chapter 3, is a promising technique with reduced local network usage while still maintaining a relatively high performance in some situations. However, TMRC has some drawbacks. First, in the weak signal environment such as strong fading across all cooperative nodes, TMRC works less effectively. In an extreme case, it can cause the cooperation to stop as no node has the signal above the threshold $\gamma_T$. Secondly, its local bandwidth is not strictly bounded by the threshold $\gamma_T$. If all the long-range
links are good and not experiencing fading, every node is transmitting signal to the master. This not only wastes the local bandwidth but also possibly overflows the local wireless network.

One solution is to have an automatic mechanism to dynamically adjust the threshold. If all the nodes are in the strong fading environment for their long-range links, it should lower the threshold to allow cooperation and obtain the energy gain. If all the nodes are having good long-range links, it should raise the threshold to reduce the unnecessary local communication. However, this dynamic control mechanism can be difficult to implement. One problem is that if the channel environment changes rapidly, it could cause the threshold to oscillate and downgrade the performance. To solve these issues, we propose a different approach – Priority Maximum-Ratio Combining.

Consider a system of one base station and \( M \) cooperative mobile stations including a master node. For every packet (or time slot) PMRC consists of identifying the \( N - 1 \) strongest signals out of the \( M - 1 \) cooperating neighbours, relaying their sampled signals to the master, and combining with the signal received by the master node (destination) before demodulation. At the master node, MRC is used to combine the selected signals from multiple sources.

In the following we denote by \((M, N)\)-PMRC a scheme where \( M \) nodes are in cooperation and the master’s signal is combined with the signal from \( N - 1 \) cooperating neighbors. The non-cooperative mode is therefore identical to \((M, 1)\)-PMRC and the traditional \( M \)-MRC (MRC with \( M \) branches) is identical to \((M, M)\)-PMRC. We will show that \((M, N < M)\)-PMRC (e.g. \( M = 5, N = 3 \)) are the most interesting schemes that benefit from dis-
tributed diversity at low and bounded bandwidth/energy cost.

In order to study the performance of PMRC, we first study the distribution of the combined SNR. This allows us to compute the outage probability and also the BER of the combined signal. The BER allows us to derive the Frame Error Rate (FER) and finally the system throughput.

We analyse the SNR distribution of PMRC in two steps. First, we derive the probability distribution function of the combined SNR of the \(N-1\) strongest assisting nodes (which we call \((M,N)-\text{SPMRC})\), and then calculate the probability distribution of the SNR of the master node combined with the \(N-1\) strongest signal nodes.

### 4.2 SPMRC: SNR Distribution for \(N = 1, 2, 3\)

Let \((M,N)-\text{SPMRC}\), be the combined signal of the \(N\) strongest assisting nodes excluding the master node. To derive the SNR probability distribution of PMRC, as a first step we derive the probability distribution of SPMRC. The combining of these \(N\) signals is based on the traditional MRC technique.

In the case of \((M,1)-\text{SPMRC}\), the master uses the strongest signal of the \(M\) neighboring nodes. This is known as Selective Combining [62].

**Theorem 4.2.1.** Let \(\gamma_s\) be the SNR of the output signal of \((M,1)-\text{SPMRC}\) under i.i.d. Rayleigh fading. The probability distribution of \(\gamma_s\) is

\[
p_{\gamma_s}(\gamma) = \frac{M}{\gamma} e^{-\frac{\gamma}{\theta}} (1 - e^{-\frac{\gamma}{\theta}})^{M-1} \tag{4.1}
\]

**Proof.** Let \(X\) denote the random variable for the highest SNR among all \(M\)
cooperative neighbors.

\[ p_X(x) = p(\text{one node has } \gamma = x) \times p(\text{all the rest } M - 1 \text{ nodes have } \gamma \leq x) \]

\[ p_X(x) = \frac{M}{\gamma} e^{-\frac{x}{\gamma}} \left( 1 - e^{-\frac{x}{\gamma}} \right)^{M-1} \]

Because the master uses the strongest signal of all the \( M \) neighboring nodes, we have

\[ p_{\gamma_S}(\gamma) = p_X(x) \]

\[ \square \]

In the case of \((M, 2)\)-SPMRC, the master collects the two strongest signals from the \( M \) neighboring nodes.

**Theorem 4.2.2.** Let \( \gamma_S \) be the SNR of the output signal of \((M, 2)\)-SPMRC under i.i.d. Rayleigh fading. The probability distribution of \( \gamma_S \) is

\[ p_{\gamma_S}(\gamma) = \frac{M(M-1)e^{-\frac{\gamma}{\bar{\gamma}}}}{2\bar{\gamma}} \left( \frac{\gamma}{2\bar{\gamma}} + \sum_{i=1}^{M-2} \frac{(-1)^i (M-2) \left( 1 - e^{-\frac{\gamma}{2\bar{\gamma}}} i \right)}{i} \right) \] (4.2)

**Proof.** Let \( X \) denote the random variable for the highest SNR among the \( M \) neighboring nodes, and \( Y \) denote the random variable for the second highest SNR. The joint probability density function for the two random variables is

\[ p_{X,Y}(x, y) = \begin{cases} \frac{M}{\gamma} e^{-\frac{x}{\gamma}} (M-1) e^{-\frac{y}{\gamma}} \times & x \geq y \\ (1 - e^{-\frac{y}{\gamma}})^{M-2}, & x \geq y \\ 0, & \text{otherwise} \end{cases} \]
4.2. SPMRC: SNR DISTRIBUTION FOR $N = 1, 2, 3$

Applying MRC to the two strongest signals $X$ and $Y$ gives $\gamma_\Sigma = X + Y$ [21], so the probability distribution of the SNR of the combined signal is

$$p_{\gamma_\Sigma}(\gamma) = \int_0^\gamma p_{X,Y}(\gamma - y, y) \, dy$$

$$= \int_0^{\frac{\gamma}{2}} M(M - 1)(\frac{1}{\gamma})^2 e^{-\frac{\gamma}{2}}(1 - e^{-\frac{\gamma}{2}})^{M-2} \, dy$$

$$= \frac{M(M - 1)e^{-\frac{\gamma}{2}}}{\frac{\gamma}{2}} \times$$

$$\left( \frac{\gamma}{2^{\gamma}} + \sum_{i=1}^{M-2} (-1)^i \frac{1}{i} \frac{(M-2)!}{i!} (1 - e^{-\frac{\gamma}{2}}) \right)$$

\[\square\]

In the case of $(M, 3)$-SPMRC, the master collects the three strongest signals from the $M$ neighboring nodes.

**Theorem 4.2.3.** Let $\gamma_\Sigma$ be the SNR of the output signal of $(M, 3)$-SPMRC under i.i.d. Rayleigh fading. The probability distribution of $\gamma_\Sigma$ is

$$p_{\gamma_\Sigma}(\gamma) = \frac{1}{2} M(M - 1)(M - 2)(\frac{\gamma}{\gamma})^3 e^{-\frac{\gamma}{2}} \times$$

$$\left( \frac{\gamma^2}{6} + \sum_{i=1}^{M-3} (-1)^i \frac{(M-3)!}{i!} \times$$

$$\left( (1 - e^{-\frac{\gamma}{2}})(\gamma e^{-\frac{\gamma}{2}} - \frac{3\gamma^2}{4}) + \gamma e^{-\frac{\gamma}{2}} \right) \right)$$

\[4.3\]

**Proof.** Let $X$ be the random variable for the highest SNR in $M$ nodes, $Y$ be the random variable for the second highest SNR, and $Z$ be the random variable for the third highest SNR. The joint probability density function of the three random variables is
\[ p_{X,Y,Z}(x, y, z) = \begin{cases} \frac{M}{\gamma} e^{-\frac{x}{\gamma}} (M-1) \frac{1}{\gamma} e^{-\frac{y}{\gamma}} (M-2) \frac{1}{\gamma} e^{-\frac{z}{\gamma}} \\ (1 - e^{-\frac{x}{\gamma}})^{M-3}, \quad x \geq y \geq z \\ 0, \quad \text{else} \end{cases} \]

According to MRC, \( \gamma_{\Sigma} = X + Y + Z \) \[21\]. Therefore, the probability distribution of the SNR of the combined signal is

\[
p_{\gamma_{\Sigma}}(\gamma) = \int_{D_{y,z}} p_{X,Y,Z}(\gamma - y - z, y, z) \, dy \, dz
\]

\[
= \int_{0}^{\frac{\gamma_{\Sigma}}{2}} \int_{0}^{\gamma_{\Sigma} - y} p_{X,Y,Z}(\gamma - y - z, y, z) \, dy \, dz
\]

\[
= \frac{1}{2} \frac{1}{M(M-1)(M-2)} \frac{1}{\gamma} e^{-\frac{\gamma}{\gamma}} \times
\]

\[
\left( \frac{\gamma^2}{6} + \sum_{i=1}^{M-3} (-1)^i \frac{(M-3)}{i} \right) \times
\]

\[
\left( (1 - e^{-\frac{\gamma}{\gamma}})(\gamma - \frac{3\gamma^2}{i} + \gamma e^{-\frac{\gamma}{\gamma}}) \right)
\]

\[\blacksquare\]

### 4.3 PMRC: SNR Distribution for \( N = 2, 3, 4 \)

In PMRC, the master node always combines its own received signal with the \( N - 1 \) strongest signals from the assisting nodes. Using its own signal does not incur any local bandwidth usage or energy consumption and always improves the combined SNR. In Section 4.2 we have derived the SNR distribution of the combined signal of the \( N - 1 \) strongest signals from the assisting nodes (Equation 4.1, 4.2, and 4.2). So the SNR distribution of PMRC can be described in the following theorem.
4.4. PERFORMANCE ANALYSIS

Theorem 4.3.1. Let $\gamma_{\Sigma'}$ denote the random variable of the PMRC signal’s SNR at the master node, $\gamma_{\Sigma}$ be the random variable of the combined signal SNR from $N - 1$ assisting nodes, and $\gamma$ be the random variable of the signal SNR of the master node. The probability distribution of $\gamma_{\Sigma'}$ is

$$p_{\gamma_{\Sigma'}}(\gamma) = \int_0^\gamma p_{\gamma_{\Sigma}}(\tau) \cdot p_{\gamma}(\gamma - \tau) d\tau$$  \hspace{1cm} (4.4)

Proof. Because the master combines the signal from assisting nodes and the signals from itself using MRC, we have

$$\gamma_{\Sigma'} = \gamma_{\Sigma} + \gamma$$

Since the probability distribution of the sum of two independent random variables is the convolution of the two random variables, we obtain

$$p_{\gamma_{\Sigma'}}(\gamma) = \int_0^\gamma p_{\gamma_{\Sigma}}(\tau) \cdot p_{\gamma}(\gamma - \tau) d\tau$$

\[\square\]

Computing the SNR probability distribution for higher values of $N$ is harder to obtain analytically in a closed form formula. However, we will show that small values of $N$ are sufficient to obtain most of the diversity gain.

4.4 Performance Analysis

The performance of PMRC can be tuned by the system parameters $M$ and $N$. If $N = 1$, it is the case of no cooperation as no assisting node sends its contribution. If $N = M$, it is the case of full diversity $M$-node MRC case. So the
performance of PMRC should be between those two cases. Unlike TMRC, where the local bandwidth usage depends on the threshold $\gamma_T$ and the average SNR $\bar{\gamma}$, the local bandwidth usage of PMRC is solely controlled by $N$. This brings the advantage that the local bandwidth is exactly determined by the system parameters rather than runtime signal levels. In this section, we discuss the trade-off between the performance and local bandwidth usage in terms of outage probability, bit error rate and throughput, and compare them with TMRC.

### 4.4.1 Outage Probability

Outage probability is a commonly used measure to evaluate the performance of communication systems. As initially presented in Section 1.3.1, let $\gamma_0$ be the minimum SNR for acceptable performance of the demodulator. The outage probability is defined as

$$P_{out} = p_{\gamma_{\Sigma'}}(\gamma \leq \gamma_0) = \int_{0}^{\gamma_0} p_{\gamma_{\Sigma'}}(\gamma) d\gamma,$$

where $p_{\gamma_{\Sigma'}}$ is derived in Theorem 4.3.1.

Figure 4.1 shows the performance of $(5, N)$-PMRC for $N = 2, 3$ and 4, and compares it to the non-cooperative scheme and the traditional MRC. For the same outage probability $(5, 2)$-PMRC results in a significant energy saving (i.e., the value of required $\bar{\gamma}$) against non-cooperative scheme. For example, for a target $P_{out} = 10^{-2}$, the average transmission energy can be reduced by more than 17dB, which is 50 times less energy. Even higher saving can be achieved for lower values of $P_{out}$. One can note that most of the diversity gain is already achieved using $(5, 3)$-PMRC in comparison with the traditional MRC. From this graph, we conclude that most of the benefits
of the traditional MRC can be achieved by requesting the samples from a few neighbors as long as we focus on the strongest signals. Therefore, PMRC can accommodate the limited short-range bandwidth, and still provide most benefits of MRC.

![Figure 4.1: Outage Probability of PMRC (N = 2,...,4) vs. MRC and non-cooperative case.](image)

![Figure 4.2: Impact of M on the performance of PMRC.](image)

We also studied the impact of the number of cooperating nodes on the outage probability. Figure 4.2 shows that increasing $M$ tremendously reduces the outage probability. For example, although 5-MRC outperforms (5, 3)-PMRC, increasing $M$ by 1 gives (6, 3)-PMRC, which not only outperforms 5-PMRC (by 2dB at $P_{out} = 10^{-7}$) but also requires only 2 cooperating nodes to send their contribution instead of 4 nodes as is the case of 5-MRC. Therefore 5-MRC requires 100% more bandwidth for lesser performance than (6, 3)-PMRC. A similar improvement in outage probability and bandwidth requirement can be observed when comparing 7-MRC to (7, 4)-PMRC and (8, 4)-PMRC. 7-MRC requires 6 cooperating nodes to send their contribution while (8, 4)-PMRC only requires the top three to contribute. From the many comparisons we carried, we observe that as the average SNR
increases, \((M', 1)\)-PMRC will eventually outperform \(M\)-MRC or any \((M, N)\)-PMRC if \(M' > M\).

### 4.4.2 Bit Error Rate

We consider the same environment as the one we used to calculate the bit error rate of TMRC. With the probability distribution of the SNR of PMRC derived in Theorem 4.3.1, we can compute the bit error rate of the PMRC system in the coherent Minimum-Shift Keying (MSK) modulation with uncoded communication.

![Figure 4.3: Bit Error Rate of coherent MSK demodulator under PMRC \((N = 2, ..., 4)\) vs. MRC and non-cooperative case.](image)

![Figure 4.4: Impact of \(M\) on the performance of PMRC in terms of Bit Error Rate.](image)

We first compare the performance of PMRC to the non-cooperative mode and the traditional MRC. The BER performance of PMRC is consistent with the outage probability. Figure 4.3 shows that for a target BER of \(10^{-3}\), \((5, 2)\)-PMRC requires 20dB (100 times) less power and with the contribution from only one cooperating neighbor. Higher gains are achievable when the target BER is lower. For the values of interest to wireless communication engineers,
most of the gain of MRC is obtained using the 2 to 3 strongest neighboring signals. Figure 4.4, shows the impact of increasing $M$. Similar to the outage probability, increasing the number of cooperating nodes outweighs the benefit of increasing $N$ the number of nodes who are actually sending their contribution.

**Figure 4.5:** Bit error rate: PMRC vs TMRC ($M=5$).

**Figure 4.6:** Local bandwidth usage of $(5, 2)$-PMRC and TMRC ($\gamma_T = 0 dB, M = 5$).

PMRC and TMRC are both Pre-Demodulation Combining techniques. In TMRC, $\gamma_T$ is used to control the amount of local communication, but the actual amount also depends on the SNR of each assisting nodes. In PMRC, the amount of local communication solely depends on the parameter $N$. Because of this property, PMRC is a more preferable choice for Pre-Demodulation Combining. Figure 4.5 shows the bit error rate comparison between PMRC and TMRC under various parameter settings. For the case of 5 nodes in cooperation, we observe that TMRC with $\gamma_T = 0$ has a higher BER than $(5, 2)$-PMRC when the long-range signal quality is low. As the signal quality rises the BER of the TMRC with $\gamma_T = 0$ eventually drops below the BER of $(5, 2)$-PMRC but within a small difference. However, in
Figure 4.6, we find that TMRC with $\gamma_T = 0$ uses significantly more local bandwidth when the long-range signal quality is strong while PMRC strictly limits the local bandwidth usage with the parameter $N$.

### 4.4.3 Throughput

The throughput of PMRC can be computed as described in Section 1.3.1. In the same way of analysing TMRC, we consider an overhead of CRC (32 bits in this case), and use the packet size that maximizes the normalized throughput.

**Figure 4.7:** Throughput of PMRC with various $E_b/N_0$(dB) and the number of nodes in cooperation.
4.4. PERFORMANCE ANALYSIS

Our analysis shows that the throughput of the master node can be tremendously increased by signal combining with a limited number of cooperative nodes (See Figure 4.7). We also find that PMRC gives comparable performance of MRC by using fewer active branches. For example, for $N = 3$ and $M = 5$ (Figure 4.7c), besides the master’s branch it uses only two active branches out of the four external diversity branches, but it still achieves a throughput of more than 0.9 with a fairly low $E_b/N_0$ $(4\,dB$ or above). This tells us that its throughput performance must be at least 90% of the performance given by MRC (the maximum is 1) while it uses only half of the bandwidth required by MRC. With a very low $E_b/N_0$ at value 1, it still maintains the throughput at 0.65.

4.4.4 Local Bandwidth Usage

PMRC guarantees its local bandwidth usage. For direct use of $(M, N)$-PMRC, the local bandwidth usage is always a constant $N - 1$. PMRC is implemented as a Pre-Demodulation Combining technique in HPC. We now compute the local bandwidth usage of the HPC which uses PMRC.

We consider a simplified two-level HPC strategy which consists of Decode-and-Forward Combining and Pre-Demodulation Combining. The local bandwidth can be computed by considering three cases. First, the master can correctly decode the frame/packet (this occurs with probability $1 - FER$). In the second case, the master is unable to decode the packet but at least one of the $M - 1$ assisting nodes is capable of decoding it (this occurs with probability $FER * (1 - FER^{M-1})$). Finally, in the third case PMRC will be done and the $N - 1$ assisting nodes with the strongest signals have to send their sampled signals (this happens with probability $FER^M$). Thus,
the average local bandwidth requirement is

\[
\text{Avg} - \text{Throughput}_{\text{short-range}} = FER \times (1 - FER^M) + (N - 1) \times R \times FER^M,
\]

where \( R \) denotes the number of bits required from each assisting node for the combining. \( R \) can be equal to 8 in the case of a non-coherent decoding or soft decision combining, and as high as 96 for coherent decoding (4 oversampling factor, 12 bits quantization for \( I \) and \( Q \)).

\[\begin{array}{c}
\text{Figure 4.8:} \text{ Bit error rate and local bandwidth usage of } (5, 4)\text{-PMRC based HPC. (Packet Length is 1500 bytes)}
\end{array}\]

Assume that transmitting the sampled signal can be significantly larger than the decoded signal, we can use the PMRC local traffic as the approximation of the total local traffic. Figure 4.8 demonstrates the bit error rate and the local bandwidth usage of \((5, 4)\)-PMRC based HPC against different signal quality values. As we can see, if the signal quality is low (such as \( \bar{\gamma} = -1dB \)), then both the bit error rate and the local bandwidth usage are high, because the 3 assisting nodes are helping the master node all the time; As the signal quality rises the bit error rate drops, and due to the HPC strat-
egy the local bandwidth usage is reduced even more sharply (nearly zero at 10dB).

4.5 Protocol Implementation for PMRC

**Algorithm 3: PMRC - Master Node Protocol**

Initialize the cooperative network with \( M \) nodes

Broadcast the cooperation control packet - CCINFO

/* CCINFO contains the info. such as frequency, modulation, GSM time slot allocation, and parameters \((M, N)\) for the PMRC cooperation scheme. */

begin

while until the session ends do

\( \text{buf}[0] \leftarrow \text{receive signal at the next expected time slot from the long-range interface; } \)

\( \Gamma[0] \leftarrow \text{the SNR } \gamma \text{ of the received signal; } \)

\( \Delta[0] \leftarrow 1 \text{ if } \gamma > \gamma_D \text{ and CRC correct; } \)

if \( \Delta[0] = 1 \) then

\( \text{Broadcast to cancel cooperation; } \)

\( \text{out} \leftarrow \text{decode}(\text{buf}[0]); \text{ return; } \)

Broadcast the Phase I beacon to all nodes through the short-range interface;

\( \Gamma[1...M] \leftarrow \text{collect } \gamma \text{ from all branches; } \)

\( \Delta[1...M] \leftarrow \text{collect } \delta \text{ from all branches; } \)

if \( \text{sum}(\Delta[1...M]) > 1 \) then

\( \text{out} \leftarrow \text{Received data at Phase II; return. } \)

if \( \text{num}(\gamma' > \gamma_D) \geq S, \gamma' \in \Gamma[1...M] \) then

\( \text{out} \leftarrow \text{soft decision decode on the aggregated data from N strongest neighboring nodes; return; } \)

\( \text{buf}[1...N - 1] \leftarrow \text{collect the sampled signals with the top } (N - 1) \text{ SNRs from assisting nodes; } \)

\( \text{out} \leftarrow \text{decode}(\text{MRC(buf,}\gamma)); \)

/* \text{out is the output data } */

end
Algorithm 4: PMRC - Assisting Nodes Protocol

Receive the cooperation control packet - CCINFO.

begin

while until the session ends do

buf ← receive signal at the next master’s time slot from the long-range interface;

γ ← the SNR of the received signal;

δ ← the result from CRC check if γ > γ_D;

Wait for the Phase I beacon from the master;

Receive the Γ[1...M] and ∆[1...M] from all other assisting nodes;

Broadcast γ and δ at its dedicated time slot;

Wait for the Phase II beacon from the master;

if it’s the highest ID with δ = 1 then

Send the decoded packet to master; return;

if other nodes pass the CRC check then return;

if γ > γ_D and num(γ’ > γ_D) ≥ S, γ’ ∈ Γ[] and γ is among the N strongest signal branches then

send soft decision decoding values to the master

return;

if γ is within Nth strongest SNR of all assisting nodes then

transmit buf to the master in the i_th time slot through the short-range interface.

end

Algorithm 3 and Algorithm 4 outline the PMRC based two-level HPC protocol implementation. This protocol consists of two phases. In the first phase, all the nodes broadcast their received signal quality, and decide what strategy to use and which assisting nodes are selected. In the second phase, the selected nodes transmit their contribution to the master node and combine with the master’s signal.
4.6 Simulations

In Section 4.4.2, we show that PMRC can significantly reduce the BER. Like many other combining techniques, it assumes perfect phase synchronization among the signal branches. However, in practice it is well known to be a challenging task. To solve this problem, we use a pilot-based technique for estimating the channel condition and adjusting the phase of each signals. Our simulation testbed is created using Matlab and Simulink.

![Graph showing BER performance of PMRC with pilot-based channel estimation.](image)

**Figure 4.9:** BER performance of PMRC with pilot-based channel estimation.

We consider a long range communication link using the Gaussian Minimum Shift Keying modulation (such as used in the GSM cellular communication standard) with a 200 kHz frequency band, and a symbol rate of 250 ksp.s. For channel estimation, we supplement the data signals with a pilot tone (i.e., non modulated sin wave) separated by 200 kHz from the center of the communication band. Similar techniques are commonly used in many communication systems such as IEEE802.11a (4 pilots for 48 carriers), and
WiMax (8 pilots for 256 carriers). Note that a single pilot tone can be shared by multiple frequency bands. On the receiver side, the pilot tone is filtered and used to estimate the channel to resynchronize the master and assisting nodes signals before combining. We use an over-sampling rate of 4 samples per bit.

Figure 4.9, summarizes the performance of PMRC for $M = 5$. The simulation results confirm that significant gains can be achieved by combining the master data with the two strongest assisting nodes. A gain of 10 dB (order of magnitude reduction in energy cost) is reachable. We also note that due to the imperfection of the phase synchronization technique a full MRC has poorer results than $(5, 4)$-PMRC, so it is preferable to only combining with the strongest signals.
In Chapter 3 and Chapter 4, we introduced two pre-demodulation signal-combining techniques. They are effective methods to overcome channel fading, but transmitting the sampled signals takes significant local bandwidth and can be too costly for the energy limited mobile handheld devices. In this chapter, we introduce a Post Soft-Demodulation Combining technique as part of the HPC strategy. We evaluate its performance in terms of bit error rate, throughput and local bandwidth usage, and compare them with TMRC and PMRC. In the end, we present our experimental results from our implementation prototype on a software-defined radio platform.

5.1 Introduction

Pre-Demodulation Combining techniques such as TMRC and PMRC can generate a significant amount of local traffic. The signal is first down-converted to the intermediate frequency and then sampled using an analog-to-digital converter (ADC). However, this sampled signal can be substantially larger
than the information it contains. For example, it can take 96 bits per symbol (4 over-sampling factor, 12 bits quantization for $I$ and $Q$). Therefore, directly transmitting the sampled signal is not very efficient and should be avoided if possible.

A simple solution is to completely decode the packet and forward the decoded packet to the master. This method produces the minimum local bandwidth footprint, and actually it is the Decode-and-Forward technique used in our HPC strategy. It works fine if the packet can be correctly decoded independently. This can be the case when a mobile node is experiencing strong uneven fading or shadowing. However, it is less effective for correcting errors. For instance, in the case of $N = 2$ if one node gets bit 0 and the other node gets bit 1, the combiner has no way to determine which one to use. So it has no performance improvement. In the case of $N \geq 3$, it is equivalent to repetition codes which are known to have a poor performance for error correction.

A better solution is to use the soft-decision values from the demodulator instead of the hard values. A soft-decision value $SV$ is a real number in $[-1, 1]$. In the case of binary, if $SV < 0$ it represents 0, otherwise 1. It means the confidence or how close of being 0 or 1. Due to the extra information they provide, they can have a better error correction ability in comparison with the hard values. In our analysis, we take $SV$ as probability approximately. Soft-decision values can be encoded in very few bits. In Section 5.5.2, we will show each value can be compressed to as few as 3 bits with a small performance compromise. Therefore, transmitting them would cause much lighter local bandwidth usage than transmitting the sampled signal.
5.2 Maximum Likelihood Soft Combining

Upon receiving a set of soft-decision values from the assisting nodes, the master node needs a method to combine those values and the value from itself, and output the most likely initially transmitted value. One simple solution is to take the value which has the highest confidence. Another simple solution is to take a majority vote or the sum of all the soft values. However, these are suboptimal combining techniques.

In this chapter, we will investigate the following questions: What is the optimal technique for soft-decision value combining? What is the performance compared to the Pre-Demodulation Combining techniques? And how much local communication traffic is generated? With these questions in mind, we introduce Post Soft-Demodulation Combining (PSDC). Similar to the Pre-Demodulation Combining setup, the system consists of $M$ cooperative nodes. PSDC uses a subset of $N - 1$ nodes with the strongest signal levels among all assisting nodes and combines them with the signal from the master node. It uses Maximum Likelihood Soft Combining algorithm to combine the soft values from multiple signal sources. In Section 5.2 we will show that the Maximum Likelihood Soft Combining algorithm produces the value with the lowest error probability.

5.2 Maximum Likelihood Soft Combining

First, we need to transform the soft-decision values into a form that can be used by the combiner. For a given soft-decision value $SV$ (float), in the case of binary it is 1 if $SV \geq 0$ and 0 if $SV < 0$. We can transform each $SV$ to a hard value $y$ associated with an error probability $Pe$. Inspired by the Maximum-Likelihood receiver [60], our combining technique - Maximum-
Likelihood Soft Combining is as follows:

Let \( y_i \) be the decoded hard decision value (0 or 1 for the binary case) from the \( i^{th} \) signal source. \( \vec{Y} = (y_1, ..., y_n) \) represents a vector of hard decision values from the \( n \) signal sources. Let \( P_{e_i} \) be the error probability for value \( y_i \), and \( \vec{P}_e = (P_{e_1}, ..., P_{e_n}) \) represents a vector of error probabilities for the vector \( \vec{Y} \). We also assume that all the branches are the independent, which is a common assumption in fading environments where the receivers are well separated. As a result, the errors from different nodes are independent. \( \vec{e} = (e_1, ..., e_n) \) represents a vector of errors.

\[
Pr(\vec{e}) = Pr(e_1) \times \ldots \times Pr(e_n)
\]

For a \( u \)-ary system, the Maximum-Likelihood Soft Combining decoder combines the \( n \) signal sources to produce an outcome that is the most probable (Equation 5.1). Therefore, it minimizes the bit error rate.

\[
\hat{x} = \arg \max_{0 \leq x \leq u} Pr(X = x | \vec{Y} = (y_1, ..., y_n))
\]  

(5.1)

In the case of a binary channel (\( u = 2 \)), the Maximum-Likelihood Soft Combining algorithm works as follows: For a given input \( \vec{Y} \) and \( \vec{P}_e \), the decoder runs the decision function (Equation 5.2), and outputs its result.

The decision function is defined as

\[
MLSC(\vec{Y}, \vec{P}_e) = MLSC((y_1, ..., y_n), (P_{e_1}, ..., P_{e_n})) = \begin{cases} 0, & \prod_{k=1}^{n} \left( \frac{P_{e_k}}{1 - P_{e_k}} \right)^{-y_k} \leq 1 \\ 1, & \text{otherwise} \end{cases}
\]  

(5.2)

**Theorem 5.2.1.** Let us assume that the source sends 0s and 1s with the same probability, \( Pr(X = 0) = Pr(X = 1) = \frac{1}{2} \) (if not the data can be com-
pressed). Given the received bit vector $\vec{Y} = (y_1, \ldots, y_n)$ and error probability vector $\vec{Pe} = (Pe_1, \ldots, Pe_n)$ from the $n$ signal branches, the Maximum-Likelihood Soft Combining produces the most probable value.

**Proof.** Let $X$ be the random variable for the output resulting from the MLSC function. Here we calculate $Pr(X|\vec{Y})$. Without loss of generality, we can first compute the probability of $X = 0$, given a vector of $\vec{Y}$ and an associated error probability vector $\vec{Pe}$. We then compute the probability of $X = 1$ under the same condition. Finally, we compare of those two values to verify the theorem.

The probability of $X = 0$ is

$$Pr(X = 0|\vec{Y} = (y_1, \ldots, y_n)) = \frac{Pr(X = 0, \vec{Y} = (y_1, \ldots, y_n))}{Pr(\vec{Y} = (y_1, \ldots, y_n))} = \frac{Pr(X = 0, \vec{X} + \vec{\varepsilon} = (y_1, \ldots, y_n))}{Pr(\vec{X} + \vec{\varepsilon} = (y_1, \ldots, y_n))} = \frac{\sum_{x=0}^{1} Pr(\vec{X} + \vec{\varepsilon} = (y_1, \ldots, y_n)|X = x)Pr(X = x)}{Pr(X = 0, \vec{\varepsilon} = (y_1, \ldots, y_n))} = \frac{1}{\sum_{x=0}^{1} Pr(\vec{X} = (y_1, \ldots, y_n), X = x)} = \frac{1}{1 + \prod_{k=1}^{n} \left(\frac{Pe_k}{1-Pe_k}\right)^{-y_k}}$$

The probability of $X = 1$ can be calculated as

$$Pr(X = 1|\vec{Y}) = 1 - Pr(X = 0|\vec{Y})$$

According to the Maximum-Likelihood Soft Combining algorithm (Equation 5.2), if $Pr(X = 0|\vec{Y}) \geq Pr(X = 1|\vec{Y})$, its outcome is 0, otherwise 1. We
have

\[ Pr(X = 0|\vec{Y}) \geq Pr(X = 1|\vec{Y}) \]

\[ \Rightarrow \]

\[ Pr(X = 0|\vec{Y}) \geq 1 - Pr(X = 0|\vec{Y}) \]

\[ \Rightarrow \]

\[ Pr(X = 0|\vec{Y}) \geq \frac{1}{2} \]

\[ \Rightarrow \]

\[ \prod_{k=1}^{n} \left( \frac{P_{e_k}}{1 - P_{e_k}} \right)^{(-y_k)} \leq 1 \]

\[ \Box \]

5.3 SNR Probability Distribution for Priority Signal Source Vector

The proposed Post Soft-Demodulation Combining uses the soft-decision values from the master and a subset of assisting nodes with the strongest signals among all assisting nodes. Let \( M \) be the total number of nodes in cooperation including the master node. For each packet, \( N - 1 \) assisting nodes with the strongest signals transmit their soft-decision values to the master node for combining. We consider Rayleigh fading as our channel model (See Section 2.1.2).

Let \( A \) be the random variable for the SNR at the master node.

Let \( X, Y, Z \) be the random variable for the highest, the second highest and the third highest SNR among the \( M - 1 \) nodes.
In the case of $N = 2$, the joint probability of the master node and the assisting node with the highest SNR is:

$$p_{A,X}(a, x) = \frac{1}{\bar{\gamma}} e^{-\frac{a}{\bar{\gamma}}} \cdot \frac{(M-1)}{\bar{\gamma}} e^{-\frac{x}{\bar{\gamma}}} (1 - e^{-\frac{x}{\bar{\gamma}}})^{M-2}$$ \hspace{1cm} (5.3)

In the case of $N = 3$, the joint probability of the master node and the two assisting nodes with the highest SNR is:

$$p_{A,X,Y}(a, x, y) =\begin{cases} \frac{1}{\bar{\gamma}} e^{-\frac{a}{\bar{\gamma}}} \cdot \frac{M-1}{\bar{\gamma}} e^{-\frac{x}{\bar{\gamma}}} (M-2) \frac{1}{\bar{\gamma}} e^{-\frac{y}{\bar{\gamma}}} \times (1 - e^{-\frac{y}{\bar{\gamma}}})^{M-3} & \text{if } x \geq y \\ 0 & \text{otherwise} \end{cases}$$ \hspace{1cm} (5.4)

In the case of $N = 4$, the joint probability of the master node and the three assisting nodes with the highest SNR is:

$$p_{A,X,Y,Z}(a, x, y, z) = \begin{cases} \frac{(M-1)(M-2)}{\bar{\gamma}^{4}} e^{-\frac{a+x+y+z}{\bar{\gamma}}} \times (1 - e^{-\frac{z}{\bar{\gamma}}})^{M-4} & \text{if } x \geq y \geq z \\ 0 & \text{otherwise} \end{cases}$$ \hspace{1cm} (5.5)

A generalized form can be derived as follows. Let $\vec{\Gamma}$ be a vector of the random variables of the received signal SNR. Let $\gamma_i$ be the random variable for the $(i - 1)^{th}$ highest SNR among the $M - 1$ assisting nodes ($i \geq 2$). The joint probability of $i$ nodes (the master node and $i - 1$ assisting nodes with the highest SNR) is

$$Pr(\vec{\Gamma}) = Pr(\gamma_1, ..., \gamma_i) = \begin{cases} \frac{\prod_{k=1}^{i-1} (M-k)}{\bar{\gamma}^{i}} e^{-\frac{\sum_{k=1}^{i-1} \gamma_k}{\bar{\gamma}}} \times (1 - e^{-\frac{\gamma_i}{\bar{\gamma}}})^{M-i} & \gamma_2 \geq ... \geq \gamma_i \\ 0 & \text{otherwise} \end{cases}$$ \hspace{1cm} (5.6)
5.4 Performance Analysis

5.4.1 Bit Error Rate

To calculate the bit error rate, we consider the coherent Minimum-Shift Keying (MSK) modulation with uncoded communication in a Rayleigh fading channel. This is the same modulation as we used earlier to compute the bit error rate of TMRC and PMRC, so that we can compare the performance of Post Soft-Demodulation with TMRC and PMRC.

Let $P_e$ be the error probability of the value from an assisting node, and $\vec{P}_{e_{\vec{r}}}$ be the joint error probability for the vector $\vec{r}$ containing the values from multiple assisting nodes. For ease of analysis, we can assume the transmitter sends all 0’s. So $Pr(y = 0) = 1 - P_e$ and $Pr(y = 1) = P_e$. The error happens if the combiner generates a 1.

Given $\vec{Y}$ and $\vec{P}_{e}$, the bit error rate of the Maximum Likelihood Soft Combiner can be calculated as follows:

$$P_{b_{\Sigma}} = \int_0^{+\infty} \sum_{\vec{y}} MLSC(\vec{Y}, \vec{P}_{e_{\vec{r}}}) \cdot Pr(\vec{Y}) \cdot Pr(\vec{r}) d\vec{r} \quad (5.7)$$

Our analysis results demonstrate a significant BER improvement under Rayleigh fading. For a target BER of $10^{-3}$, $(5, 2)$-PSDC requires 17dB less power and with the contribution from only one cooperating neighbor (See Figure 5.1). Higher gains are achievable when the target BER is lower. We also observe most of the diversity gain can be obtained by using a few (2 to 3) strongest neighbors. Figure 5.2 shows the impact of $M$ on the BER performance. As we can see the parameter $M$ has a more dominant effect on the BER than $N$. So, it is always better, if possible, to include more
nodes as potential contributors rather than increase the number of actively cooperating nodes (who are relaying to the master).

Figure 5.1: Bit Error Rate of coherent MSK demodulator under PSDC \((N = 2, \ldots, 4)\) vs. MRC and Non-Cooperation.

Figure 5.2: Impact of \(M\) on the performance of Post-Soft Demodulation Combining in terms of Bit Error Rate.

We also compare the BER of PSDC against TMRC and PMRC. Figure 5.3 shows that the BER performance of PSDC and TMRC are very close if a proper threshold value is chosen for TMRC. Under certain conditions, TMRC performs better (See \(M = 5, \gamma_T = 0\) TMRC crosses over \((5, 4)\)-PSDC), but TMRC uses much higher local bandwidth (See Section 4.4.2). Since PSDC is similar to PMRC, and PSDC aims at reducing the local communication footprint, we want to determine how much performance loss it causes. Figure 5.4 shows that the BER of PSDC is slightly higher than PMRC under the same configuration. This can be explained by the fact that PSDC only benefits from the diversity gain; in contrast PMRC exploits both the diversity gain and the energy (antenna) gain.
5.4.2 Throughput

We can further compute the throughput using the method described in Section 1.3.1. In the same way we analysed TMRC and PMRC, we consider the packet overhead of CRC (32 bits in this case), and use the packet size that maximizes the normalized throughput.

Similar to BER, the throughput is also improved significantly (See Figure 5.5). As we can observe the largest throughput growth happens when allowing one assisting node to transmit its soft-decision values to the master. For example in the case of $M = 4, N = 2$ (Figure 5.5b), the throughput jumps from 0.11 to 0.86 if $E_b/N_0 = 10$, and from 0.01 to 0.33 if $E_b/N_0 = 4$.

Figure 5.6 and Figure 5.7 reveal the throughput difference between PSDC and PMRC. We observe that the throughput difference is smaller when $E_b/N_0$ is high but larger when $E_b/N_0$ is low. For example, if $M = 5, N = 2$ and $E_b/N_0 = 7$ the throughput difference between the PSDC and PMRC is 10%; if $M = 5, N = 2$ and $E_b/N_0 = 1$ the throughput difference is 50%. This can
be explained by the fact that if the signal quality is good, the diversity gain is far more dominant than the energy gain and both PSDC and PMRC are able to benefit most from the diversity gain; if the signal quality is bad, the energy gain rises, but PSDC is incapable of exploit the energy gain. Therefore, the PSDC should be used when the signal level is not too low, otherwise PMRC would be a better choice if the performance is critical.
5.4.3 Local Bandwidth Usage

The main benefit of the PSDC over TMRC and PMRC is that its potential to significantly reduce the local communication bandwidth usage with little performance degradation.

In comparison with Pre-Demodulation Combining which costs tens of bits to represent one symbol (i.e. 96 bits), the soft-decision values only use a few bits to represent one bit of source information. From our experiments (Section 5.5.2), with an adequate quantization the soft-decision values can be as few as 3 bits which is less than one tenth of a sampled symbol. Although it still causes several times more local traffic in comparison with the long-range communications, but it is still not a major burden over the local bandwidth assuming the local short-range wireless link (e.g., WiFi) is much faster than the long-range wireless link.

In terms of performance, if $E_b/N_0$ is not too low the proposed PSDC technique exhibits only a slight downgrade of throughput in comparison to PMRC (See Section 5.4.2). This supports the proposed HPC strategy to use
5.5. EXPERIMENTS FOR POST SOFT-DEMODULATION COMBINING

PSDC as a substitute for PMRC and TMRC if the signal level is not too low.

5.5 Experiments for Post Soft-Demodulation Combining

5.5.1 GNU Radio and USRP Testbed

GNU Radio is an open-source software-defined radio (SDR) platform and the Universal Software Radio Peripheral (USRP) is a popular hardware implementation compatible with GNU Radio [5, 19]. The purpose of software-defined radio is to bring the software as close to the radio antenna as possible. The key benefit is that we can do all the signal processing in software on a general purpose computer. This allows a tremendous flexibility during prototyping and at a relatively low cost.

GNU Radio has powerful signal processing libraries which include modulations (e.g., GMSK, PSK, QAM, etc), error-correcting codes (e.g., Reed-Solomon, Viterbi, Turbo Codes), various filters, FFT related functions, and equalizers. The basic processing unit is a block which processes streams of data from the input ports and outputs the processed result to the output ports. For efficiency the blocks are implemented in C++. The program running on the GNU Radio platform essentially connects the blocks together from a signal source (e.g. USRP) to a signal sink (e.g. a file). For fast development, GNU Radio uses Python to connect those blocks together.

The USRP board is designed specifically for the GNU Radio. The board is equipped with RF front ends (with support for various daughter boards), analog-to-digital (AD) and digital-to-analog (DA) converters, and a field-
programmable gate array (FPGA), which serves for all the functionality required from the hardware and processes the output/input data from/to the computer through a USB connection. A USRP block diagram is depicted in Figure 5.8 [19].

There are four 12 bit analog-to-digital converters (ADC). Each samples at the rate of 64M samples per second. By Nyquist theorem, it can sample a bandwidth up to 32MHz, but the effective bandwidth it can sample is smaller because of the limited throughput of the USB link (i.e., 480Mbps raw). There are four 14 bit digital-to-analog converters (DAC). They run at a rate of 128M samples per second, so in theory it can generate the signal with bandwidth as wide as 64MHz, but in practice this value is lower because of the limitations of the USB link. The FPGA chip is the central part of the USRP board. It connects all the ADCs and DACs. The FPGA chip contains digital down converters (DDC) which are used to precisely select the part of the digitized spectrum from the ADC, translate them to the baseband and decimate to allow the samples to transmit over the USB link.
5.5. EXPERIMENTS FOR POST SOFT-DEMODULATION COMBINING

On a typical receiving path, the signal is received at the antenna; first down-converted to the intermediate frequency (IF) by receiver’s RF front end; sampled by the ADC; processed in the FPGA; and finally sent to the computer over the USB link for demodulation and post processing. The transmission path is similar. The data is first coded and modulated on the computer; sent to the FPGA for interpolation through USB; passed to the DAC; up-converted to the carrier frequency by the transmitter’s RF front end and eventually radiated by the antenna.

5.5.2 Experimentation Results

We have implemented a prototype testbed of our system on the USRP/GNU Radio platform, and have measured the performance of PSDC experimentally. We use a GMSK modulation at 500 kbps on the 2.4GHz ISM band. The GMSK demodulator is modified to output the soft-decision values. We use an RF cable to connect the boards so that we can isolate our testbed from external interference on the ISM band and be able to consistently reproduce our results. A precise Rayleigh fading channel is difficult to reconstruct, so we use a software technique to emulate the Rayleigh fading effect.

We conducted experiments for a long period of time with various transmission power levels. The total amount of recorded data is over 200GB. Due to our hardware limitations we only were able to complete the BER experiments in the range from $10^{-1}$ to $10^{-6}$, but our experimental result can already show a significant improvement in BER on the current hardware. Figure 5.9 and Figure 5.10 summarize the results. For a target BER of $10^{-3}$, $(5, 2)$-PSDC requires $15dB$ less power than the non-cooperative case and with the contribution from only one cooperating node. Higher gains
are achievable when the target BER is lower. Figure 5.11 shows the system throughput derived from the BER and with the packet size 500 bytes and a 32-bit CRC overhead. As we can see, the PSDC technique is able to effectively boost the throughput in the high BER situations. Those confirm our analysis.

![Figure 5.9: Bit Error Rate of PSDC by experiments (N = 2).](image1)

![Figure 5.10: Bit Error Rate of PSDC by experiments (M = 5).](image2)

![Figure 5.11: Throughput of PSDC by experiments](image3)

Figure 5.13 shows a comparison between the experimental results of BER and the theoretic results from Section 5.4.1. We observe a gap between our

...
experimental results and the theoretic results. This phenomena can potentially be explained by the following reasons related to our testbed. The USRP hardware has limitations. From Figure 5.12, we see that the distribution of the soft-decision values on an AWGN channels is not exactly Gaussian. Furthermore, the soft-decision values generated by the current algorithm are inaccurate. The GMSK demodulator, we used in our experiments, is a quadrature demodulator.\textsuperscript{1} It calculates the angle difference by subtracting the angle of each adjacent complex sample. The soft-decision value is this angle difference multiplied by a gain, which does not exactly represent the confidence or the probability value. A better calculation algorithm for soft-decision values is needed. Finally, in the experiments, we used the GNU Radio’s built-in GMSK implementation instead of the MSK modulation that is used in our analysis. GMSK is similar to MSK except that a Gaussian filter is applied so it uses less bandwidth than MSK. The BT value of GMSK is set to 0.35 in our experiments. Nevertheless, the purpose of our experiments is to show that PSDC, as a cooperation mechanism, is feasible to implement in practical systems to improve the bit error rate.

In practice, the soft-decision values are transferred over the short-range wireless links. Although the short-range links can have much higher bandwidth, it is still necessary to minimize the amount of local traffic to be transmitted. We evaluate how quantization impacts the performance. We conducted several experiments using multiple quantization values. The results for bit error rate are shown in Figure 5.14. We find that if the soft-decision values are quantized to 6 bits, it has a performance close to a 32-bit float number (the required transmission energy is around \(0.25dB\) higher). In our opinion, a quantization of 6 bits or above is sufficient for PSDC. However,\textsuperscript{1}

\textsuperscript{1}The GNU Radio block \texttt{gr.quadrature_demod_cf}.
it still requires transferring 6 bits for 1 bit of data. We can further lower the quantization level to 3 bits. Although, the bit error rate start degrading, the result is still acceptable considering the amount of bandwidth it saves. Here we use the uniform quantization. In the future, our implementation can be further improved by using non-uniform quantization, but our current
5.5. EXPERIMENTS FOR POST SOFT-DEMODULATION COMBINING

experimental results show that the uniform quantization already reaches a good performance.
In this chapter, we explore a distributed network-layer cooperation strategy called Distributed Link-Bonding (DLB). The principle behind this strategy is to exploit the benefit from both traffic diversity and channel diversity through traffic-multiplexing, where multiple nodes pool their long-range links together to improve their effective throughput and the network efficiency. We first motivate the benefits of traffic multiplexing and link-bonding techniques; then introduce the model used in the analysis; and present the analytical and simulation results. In Chapter 7, we present several possible methods to implement DLB and our prototype implementation on some current smart phones.

6.1 Background and Motivation

Mobile phone data services are transforming the mobile world with applications such as web browsing, video/music streaming, cloud computing. Although, a set of third generation technologies have been developed and deployed for several years, the quality of service is still unsatisfactory. Users ex-
To solve these problems, instead of deploying more base stations which is the conventional approach adopted by operators to improve the service coverage, we propose to exploit user cooperation and benefit from both traffic diversity and channel diversity. We call this strategy Distributed Link-Bonding (DLB). In DLB, the cooperating nodes’ long-range links are bundled together using the local high-speed wireless network and the traffic from all nodes in cooperation are being multiplexed through this bundled link. As a result, the traffic load can be balanced among all long-range links, and therefore the average delay and bandwidth utilization can be significantly improved. It also has the advantage of being practical in terms of implementation and deployment. This method can be implemented as a software middleware on the existing hardware and be transparent to the applications.

We note some related work to the DLB approach. Link Aggregation or Multi-Link Trunking (MLT) on the link layer for the wired Ethernet is proposed in [7]. It allows to group multiple physical links into one logical Ethernet link to provide fault-tolerance and high-speed communication between routers. Multi-homed transport protocols such as IETF Stream Control Transmission Protocol (SCTP) [65] provides similar features as Link Aggregation, but does not account for the wireless environment with multiple types of air-interfaces. Furthermore, it requires the application to be aware of the network protocol to benefit from fault-tolerance and parallel links. Indirect
TCP or I-TCP [16] for mobile hosts splits the single TCP connection into two independent TCP links to prevent the performance fluctuation caused by wireless communications from propagating to the fixed network. One of the advantages of this method is that it has no change of the TCP protocol on the hosts. Our Neptune prototype for DLB implementation is inspired by this idea (See Section 7.3) and used to embed our distributed diversity strategy. Wireless Mesh Networks have also been proposed as a cooperative approach to wireless communications, and researched in [8, 11, 24]. They allow neighbors to connect their home networks together to share their Internet access via gateways that are distributed in their neighborhood. Within wireless mesh networks, a multi-radio unification protocol for IEEE 802.11 wireless networks is proposed in [13].

6.2 Model and Approach

The considered DLB system consists of $m$ cooperating nodes, each node is equipped with a long-range relatively low data-rate cellular interface (e.g., GPRS, EDGE, HSDPA, 1xEvDO), and a short-range high data-rate interface for local communications (e.g., WiFi). For our analysis, we consider the following system characteristics and parameters.

User Traffic Distribution: For our analysis and simulations, we assume that the nodes generate requests according to specific distributions. We assume that a node is blocked until its request is fulfilled. We consider a Blocked-Poisson distribution for the user traffic. A user generates requests according to a Poisson process with rate $\lambda$ but blocks until the current request is fulfilled. This distribution is motivated by the fact that mobile phones’ users
usually make one request at a time (e.g., browse a web page and wait until it is completely loaded before making a new request).

**Service Time:** This corresponds to the size of a web page, or the length of a session. We consider two types of distributions: *exponential* and *heavy-tail*. The heavy-tail distribution is commonly used to simulate the packet size/service time on the Internet, as the Internet traffic appears to have a few very long requests, which are not exponentially bounded [26, 55, 74].

**Cooperation Modes:** We consider two cases: *non-cooperative* mode and *cooperative* mode. In the *non-cooperative* mode all the traffic generated by each node/user is transmitted over the nodes’ own long-range links. In the *cooperative* mode, the traffic generated by the cooperating nodes is multiplexed over all long-range links. The combined capacity can be viewed as the sum of the capacities of all the nodes’ long-range links. In this mode, we consider two ways of multiplexing the user traffic.

- **Serial-multiplexing** consists of queuing each user/node’s request until all previous requests are serviced. This means that the request is processed at the combined rates of all long-range interfaces.

- **Parallel-multiplexing** processes all requests simultaneously. The speed of processing a request depends on the number of active requests in the system.

**Evaluation Metrics:** We study two characteristics of these systems: the average service delay for each request and the perceived throughput, which is the average throughput that a request actually experiences (See Section 1.3.1).
6.3 Stationary Regime for Blocked-Poisson Traffic, and Exponential Service Time

In this section, we analyse the system in non-cooperative mode and cooperative DLB serial-multiplexing mode using Queuing theory and a Markov-chain model. Serial-multiplexing is a simplified model, so we can derive its performance analytically. The performance of parallel-multiplexing will be evaluated through simulations (see Section 6.4).

Non-Cooperative Mode: Each node is working independently. Assume the incoming traffic is a Poisson process with arrival rate $\lambda$, and the transmission rate of the long-range link is $\mu$. And the system is running in the blocking mode, in which no new requests would be generated if current request is still transmitting. It can be modelled as a two-state Markov-chain, Figure 6.1.

![Figure 6.1: Markov chain for each node in non-cooperative mode](image)

A straightforward analysis gives the following results. Global Balance Equations: $P_1 = \rho \cdot P_0$, $\rho = \lambda / \mu$

The average number of requests in the system is: $N = \lambda / (\mu + \lambda)$

It can be easily shown that the perceived throughput is equal to $\mu$. 
**DLB Serial Multiplexing Mode:** In this part, we analyse the service delay and perceived throughput incurred by a request in the DLB serial-multiplexing mode. We assume that all the cooperative nodes have the long-range interfaces with the same characteristics (i.e., same transmission rate $\mu$). The incoming traffic from each node is a Poisson process with the same arrival rate $\lambda$. And each of the node is also running in the blocking mode. This system can be modelled as a Markov chain with $m + 1$ states (see Figure 6.2). State $i$ corresponds to the state when the system has $i$ pending requests. In the serial-multiplexing mode, only one request is processed while other requests are queued. Since the system has $m$ cooperating nodes, the processing speed is $m$ times faster in comparison to the non-cooperative mode.

Let $N_k$ denote the number of requests in the system at time $k\delta$ (including the one being processed), $P_i$ denotes the stationary probability of being in state $i$, and $P_{i,j}$ denotes the transition probability from state $i$ to state $j$.

\[
P_{i,j} = P\{N_{k+1} = j|N_k = i\}
\]

\[
P_{0,0} = P\{0 \text{ requests arrive from } m \text{ nodes}\}
\]

\[
= e^{-m\lambda\delta}
\]

\[
= 1 - m\lambda\delta + o(\delta)
\]

\[
P_{i,i\geq 1} = P\{0 \text{ arrivals from } m - i \text{ nodes, 0 departs}\}
\]

\[
= e^{-(m-i)\lambda\delta} \cdot e^{-m\mu\delta}
\]

\[
= 1 - (m - i)\lambda\delta - m\mu\delta + o(\delta)
\]
\[ P_{0,1} = P\{1 \text{ requests arrive from } m \text{ nodes}\} \]
\[= m\lambda\delta \cdot e^{-m\lambda\delta} \]
\[= m\lambda\delta + o(\delta) \]

Figure 6.2: Markov chain for \(m\) cooperating nodes in serial-multiplexing mode.

\[ P_{i,i+1} = P\{1 \text{ arrives from } m-i \text{ nodes, } 0 \text{ departs}\} \]
\[= (m-i)\lambda\delta \cdot e^{-(m-i)\lambda\delta} \cdot e^{-m\mu\delta} \]
\[= (m-i)\lambda\delta + o(\delta) \]

\[ P_{1,0} = P\{0 \text{ requests arrive from } m-1 \text{ nodes,} \]
\[1 \text{ departs}\} \]
\[= e^{-(m-1)\lambda\delta} \cdot (1 - e^{-m\mu\delta}) \]
\[= m\mu\delta + o(\delta) \]
\[ P_{i>1,i-1} = P\{0 \text{ arrives from } m-i \text{ nodes, 1 departs}\} \]
\[ = e^{-(m-i)\lambda \delta} \cdot m\mu \delta \cdot e^{-m\mu \delta} \]
\[ = m\mu \delta + o(\delta) \]

The Global Balance Equation is:
\[ P_i \cdot (m-i)\lambda = P_{i+1} \cdot m\mu \]

Let \( \rho = \frac{\lambda}{\mu} \). Then, from the balance equations:
\[ P_{i+1} = \frac{m-i}{m} \rho \cdot P_i \]
\[ P_i = \frac{(m-1)! \rho^i}{(m-i-1)! m^i} \cdot P_0 \]

Since \( \sum_{i=0}^m P_i = 1 \), we have
\[ P_0 = \left[ \sum_{i=0}^m \frac{(m-1)! \rho^i}{(m-i-1)! m^i} \right]^{-1} \]

and
\[ P_i = \left[ (m-i-1)! \left( \frac{m}{\rho} \right)^i \cdot \sum_{j=0}^m \frac{\rho^j}{(m-j-1)! m^j} \right]^{-1} \]

The average number of requests in the system is:
\[ N = \sum_{i=0}^m i \cdot P_i \]

To compute the average delay, let us consider a newly arrived request. We first compute the current system state probability: \( P(N_t = i|N_t \neq m) \), \( 0 \leq i < m \). This corresponds to the probability that \( i \) requests are already
in the system. The number of queued requests cannot be \( m \) or larger, because users are blocked after each request. Therefore, if there are already \( m \) queued requests no new request would arise.

\[
P(N_t = i|N_t \neq m) = \frac{P(N_t = i, N_t \neq m)}{P(N_t \neq m)} = \frac{P_i}{1 - P_m}
\]

For each \( P(N_t = i|N_t \neq m) \) the delay is \( X_1 + \cdots + X_i + X \), where \( X_i \) denotes the time to process (transmit) existing request \( i \) and \( X \) is the time to process the new request. The average time for processing each of these requests is \( (i + 1)/(m\mu) \). This holds true even for the request being processed because of the memoryless characteristic of the exponential distribution. Note that we cannot compute the average delay directly using Little’s Theorem, because the effective arrival rate of the system is unknown but less than \( \lambda \), which is due to the blocking nature of our request generation process. Combining all the cases, we obtain the following average delay for finishing a transmission request:

\[
T = \sum_{i=0}^{m-1} (X_1 + \cdots + X_i + X) \cdot P(N_t = i|N_t \neq m)
\]

\[
T = \sum_{i=0}^{m-1} \frac{i + 1}{m\mu} \cdot \frac{P_i}{1 - P_m}
\]

Using the same approach the perceived throughput is as the following.

\[
PT = \int_0^\infty \frac{1}{\mu} e^{-\frac{L}{\mu}} \sum_{i=0}^{m-1} \frac{L \cdot i}{(L+i)(m\mu)} \cdot \frac{P_i}{1 - P_m} dL
\]
Our simulation of the Blocked-Poisson traffic with exponential service
time and serial-multiplexing exactly matches the above analytical results
(See Figures 6.3 and 6.4). These results will be discussed the next section.

6.4 Traffic Pattern and Simulation

Our simulation testbed is built up using Matlab and Simulink. We sim-
ulate two types of traffic – Blocked-Poisson and heavy-tail in both serial-
multiplexing and parallel-multiplexing. In this section, we present the sim-
ulation results in terms of service delay and perceived throughput.

6.4.1 Blocked-Poisson Traffic (Serial vs.

Parallel-Multiplexing)

Depending on the type of applications, parallel-multiplexing might be more
practical and closer to the reality in comparison with serial-multiplexing.
However, the analysis of parallel-multiplexing is more complex, and it is
difficult to obtain a closed-form formula for it, because in such DLB system
the service rate for completing a request depends on the number of requests
within the system.

To evaluate the performance, we simulate both multiplexing schemes.
We normalize the average service time per request to 1 unit of time (i.e.,
$\mu = 1$). We vary the arrival rate for each node between $\frac{\mu}{10^3}$ and $10^3\mu$. Note
that we can use the arrival rates higher than the service time because our
arrival process is Blocked-Poisson (the length of the queue never exceeds $m$).
For each plotted point, we run the simulation for 500,000 units of time.
As reported in Figures 6.3 and 6.4, at low and moderate load both schemes show a significant reduction in delay and increase of perceived throughput. We also observe in some situation the serial-multiplexing slightly outperforms the parallel-multiplexing. This is because the service delay consists of the waiting time and the transmission time. Although in the serial-multiplexing case the request would wait if another request is under transmission, this would not slow down the request under transmission. Overall, serial-multiplexing performs better. For example, assume there are 2 requests with the same size arrive in the system, and system service rate is 1 request per unit of time. So, in serial-multiplexing, one request takes 1 unit of time, and the other request takes 2 units of time (1 unit of time for waiting and 1 unit of time for transmitting). The average is 1.5 units of time. In contrast, in parallel-multiplexing, both requests takes 2 units of time. However, we note that serial-multiplexing exhibits a higher variance of service delay (jitter).

\[ \lambda / \mu, \mu = 1 \]

**Figure 6.3:** Average delay per packet for serial and parallel-multiplexing with Blocked-Poisson traffic.
6.4.2 Heavy-Tail Traffic

The distribution of the size of web pages is commonly modelled as more of a heavy-tail than an exponential distribution [59, 26, 36]. Because the analysis of the cooperative mode is much harder for this type of traffic, we also use simulations to quantify the performance improvement resulting from DLB. One popular distribution for modelling heavy-tail traffic is the Pareto distribution [55, 74] (See Section 1.2.1.3 for definition).

Figure 6.5 indicates that under heavy-tail traffic, significant increase of the perceived throughput can be achieved when the load is low or moderate. We also observe that the serial-multiplexing outperforms the parallel-multiplexing, and in some situations the difference is substantial. For example, the parallel-multiplexing results in up to 45% lower perceived throughput than the serial-multiplexing compared to a 15% in exponential traffic, see Figure 6.6 and Figure 6.7.

We also compared the performance of DLB under heavy-tail and expo-
Figure 6.6: Ratio of perceived throughput of parallel to serial-multiplexing for Blocked-Poisson traffic.

Figure 6.7: Ratio of perceived throughput of parallel to serial-multiplexing for heavy-tail traffic.

dential traffic. Figures 6.8 and 6.9 show the performance of serial-multiplexing DLB under both traffic for the same average request length. In some situation such as moderate load, heavy-tail traffic results in a perceived throughput up to 80% better than exponential traffic. This is because for the heavy-tailed traffic there are many short requests, but also a few very long requests. In contrast, the exponential traffic with the same mean would have a far more medium size requests. And transmitting long requests is more efficient than short requests, given the same total amount of data. For example, assume there are 2 requests each with 1 unit of size arriving at the system, and the system transmit rate is 1 unit of size per unit of time. So one request takes 1 unit of time, and the other request takes 2 units of time (1 unit of time for waiting and 1 unit of time for transmitting). So they together take 3 units of time. In contrast, assume one long request with 2 units of size, which is the same amount of data as the previous case, it would take only 2 units of time to complete.
6.5 Rate Allocation

Rate allocation is important to the DLB system, because it decides how the packets from the traffic source are scheduled and routed through the cooperative network. Failing to have such a rate allocation algorithm, which accounts for the network condition, would cause the local network to congest on some links while underuse other links. Thus, it degrades the overall system performance.

Fortunately, this problem can be easily addressed in our cooperative environment. This is because the proposed DLB system has a simple topology that there is only one hop between cooperative nodes through the local high-speed network. Besides, the two types of air-interface used in the DLB systems are orthogonal, so the local wireless communications would not interfere with the long-range wireless communications.
The strategy we adopt in the DLB system is a water-filling algorithm (or local queue balancing). For each cooperative node the long-range communication queue status, such as length and delay, is monitored and reported to the master node. When the traffic from the application layer arrives at the DLB packet dispatcher of the master node, it measures the queue status for each assisting nodes and dispatch the packet to the assisting node with the best queue condition such as minimum queue length or shortest queue delay. For ease of implementation, our prototypes adopt the queue length as the metric.

This rate allocation algorithm has the benefit that the channel condition can be calculated quickly and accurately as the queue length or delay is an effective measure of the channel condition. Besides, the algorithm is relatively simple, it would not cause much overhead on the CPU or network bandwidth. However, dispatching packet with such method would not prevent the packets from reaching the destination out of order, which would cause severe damage to the performance of existing transport layer protocols such as TCP. We will address this packet re-ordering issue in detail, and propose and implement solutions during the DLB prototyping in Chapter 7.
CHAPTER 7

Distributed Link Bonding Prototyping

In this chapter, we first discuss the potential architectures and approaches for implementing DLB. We present a tunnelling based DLB prototype and the Neptune prototype. Both prototypes are implemented as system plug-ins that can run in the current network stack. We discuss the problems we have encountered during prototyping and propose solutions. Finally, we present our experimentation results and the achieved performance of DLB on smart phones.

7.1 Cooperative Solutions

In Chapter 6, we showed that the proposed DLB system can significantly improve the network efficiency and boost the perceived throughput for current cellular systems. Several schemes are possible to implement DLB, such as DLB-Aware Applications, DLB-Aware Network Layer, Multi-homed Transport Protocols, and Proxy Based Protocols. All these schemes can be implemented in software on existing hardware. However, some of these approaches necessitate the modifications of existing applications or existing Internet pro-
tocols. These are undesirable constraints. In our implementation, we built two prototypes, a - Tunneling Based Proxy and the - Neptune prototype. Both prototypes not only run on existing hardware, but also impose no changes on the current Internet. Besides, our prototypes are implemented as system plug-ins, and can be turned on and off by the user with a single click. They are transparent to the application layer, so most of the existing applications can benefit from DLB without any modifications.

We first discuss the possible schemes for implementing DLB:

- **DLB-Aware Applications**: DLB can be implemented at the application layer. Cooperation at the application layer requires modifications of the application or a complete rewrite of the application. There are many peer-to-peer cooperative applications. For example, Skype uses cooperation between nodes to improve the call quality, raise the call completion rate and traverse firewalls; PPLive uses cooperation between nodes to deliver fast online video streaming; BitTorrent uses cooperation to accelerate file download.

  In the DLB scenario, for upload traffic, whenever an application client initiates a data connection, the application client splits the data and sends it to a list of cooperating nodes. The data is forwarded to the destination application server, and finally reassembled by the application server. The download traffic flows in a similar way. However, this method is not generic and cannot be shared by all the applications. A lot of work that has to be done for each application, which makes it impractical and not scalable from a software engineering perspective.

- **DLB-Aware Network Layer**: Some modifications can be made to the network layer stack to support DLB. One way is to enable the loose
7.1. COOPERATIVE SOLUTIONS

source routing in the Internet Protocol. In the IP protocol, there are two header options - *Strict Source and Record Route* (SSRR) and *Loose Source and Record Route* (LSRR) to allow the packet sender to partially or completely specify the route that the packet takes through the network. Without specifying these options, each router on the Internet determines the routing path solely based on the packet’s destination and its own routing table. We can leverage the loose source routing option to realize the DLB cooperation.

The DLB cooperation can be implemented in the following way: When the traffic arrives at the network layer on a mobile node, for each packet a cooperating node’s IP address is added to the loose source routing record field in the IP header options. This forces the packet to travel over one of the cooperating nodes. In such a way, the traffic can be balanced over all the cooperating nodes. In the same way, when the traffic is travelling back from the application server to the mobile node, the application server also needs to add a cooperating node to the source routing record to make sure the packets go through these cooperating nodes.

Although this method looks promising, it still raises many issues. First of all, to implement this strategy, it not only requires modifications of the IP stack on the client side, but also requires the modifications of the IP stack on the application servers too. It is undesirable and unrealistic to make those changes on all the application servers throughout the Internet. Furthermore, the source routing options are usually blocked for the concerns of security attacks such as Internet address spoofing and Denial-of-Service.
• **Multi-homed Transport Protocols:** A typical multi-homed transport protocol is the IETF *Stream Control Transmission Protocol* (SCTP) [65]. Similar to UDP, SCTP is a message based transport protocol in contrast with TCP, which is a stream based protocol. SCTP provides reliable data transfer and partial ordering of the data delivery. Among many of the features that SCTP supports, multi-streaming is the one we are interested in. It allows multiple independent streams to be transmitted in parallel. For example, a web page may contain html files, javascript files, and images. SCTP allows transmitting those files simultaneously.

The DLB cooperation can be implemented using SCTP. When the traffic arrives at the transport layer (for both uploading and downloading), the DLB system starts multiple SCTP streams in a way that each one passes through a cooperating node. SCTP ensures the reliable transmission of each of the streams. On the receiver side, those streams are eventually delivered to the application layer.

One of the limitations of such a solution is that it requires all the Internet servers to run SCTP. But SCTP is not universally deployed on the current Internet. Another drawback is that to leverage parallel multi-streaming, the application needs to be aware of it. It is not completely transparent to the application. The application has to make sure each of the streams is independent and can be transferred independently. As a result, this also requires the modifications of applications.

• **Proxy Based Protocols:** In order to avoid changing the server side network stack and applications, proxies can be introduced into the system to bridge between the existing systems and the DLB cooperative environment. A diagram of the proxy based DLB platform can be seen
7.1. COOPERATIVE SOLUTIONS

in Figure 7.1.

In this approach, the system redirects the application’s uplink traffic to an Agent running on the mobile client. The Agent dispatches the data to go through each of the cooperating nodes and the destination is set to be a Remote Proxy (or DLB Proxy) on the Internet. The Remote Proxy then acts as a remote Network Address Translation (NAT) and forwards the packets to the final destination, which is an application server. For the downlink, all traffic from the application server is sent to the DLB proxy, which then sends the traffic to a set of cooperating nodes, and eventually the Agent on each cooperating nodes relays it to the original recipient.

Since all the traffic goes through this DLB proxy, a potential limitation of this method is that the proxy might become a bottleneck. Nevertheless, the advantages of this solution are promising and our prototypes adopt this proxy based approach. As the DLB cooperation gains momentum, more proxies can be deployed at the edge of the Internet and even at the server side.

Figure 7.1: Configuration of the proxy based DLB platform.
CHAPTER 7. DISTRIBUTED LINK BONDING PROTOTYPING

7.2 Tunnelling Based Architecture and Prototype

Our first prototyping attempt is a tunnelling based architecture. It uses the packet encapsulation technique commonly used in Virtual Private Networks (VPN) to create a virtual tunnel between the local cooperating nodes and the Remote Proxy. The system implementation is based on the Click Modular Router [3, 48] and runs on Linux. For the ease of implementation (in this first prototype), we use laptops with cellular PC cards as the mobile stations. The detailed software and hardware configuration can be found in Section 7.4. In the second prototype – Neptune (See Section 7.3), we are able to run the DLB system on the Google Android-based mobile phones[6]. The DLB system consists of two parts: an Agent running in each mobile node and a Remote Proxy running on a server accessible from the Internet.

The Agent is a software plug-in to the OS of the mobile node. It contains several components:

- **Status Update**: A service to broadcast and collect the identity and the status of cooperating nodes. This allows each node to discover other cooperative nodes. It also allows other nodes to learn the status information such as the pending queue length, the link quality and delay. These will be used in the packet scheduler to decide which node should be used for forwarding the traffic.

- **Data Collection**: A module to capture all the data sent from mobile applications. This is implemented as a virtual network interface bound with a virtual IP address. After changing the routing table of the mo-
7.2. TUNNELLING BASED ARCHITECTURE AND PROTOTYPE

bile node, all of the outgoing packets are sent through this interface, and therefore captured and processed by our system. This requires no modifications to the existing applications. Our implementation uses an IP over UDP encapsulation technique, which means that each IP packet is wrapped inside a UDP packet.

- **Packet Scheduler**: A module to decide which cooperating node (including itself) it should forward to, based on the rate allocation algorithm (Section 6.5). It then encapsulates the packet in a UDP packet and forwards it to the selected node.

- **Packet Forwarding**: A service on the node to receive the incoming UDP packets and forward them to the destination. Upon receiving a packet, the UDP header is removed to extract the original IP packet. If the destination of this IP packet is another cooperating node or the Remote Proxy, it encapsulates this packet in UDP again and sends it out. If the destination is itself, it delivers the packet to the application.

The Remote Proxy is a lightweight service which can be run in any Internet server. It acts like a NAT router providing packet forwarding and network address translation services.

When a UDP packet arrives at the proxy from a cooperating node, the proxy strips off the UDP header and gets the original IP packet. Then the proxy executes a network address translation (NAT) to modify the source address to its own address. Finally the packet is transmitted through the wired network and delivered to the destination. The reverse process from the destination back to the mobile node is similar. Upon receiving an IP packet from the destination, the proxy performs a NAT and changes the destination
address to the original initiator and feeds it to the packet scheduler. The scheduler then decides which cooperating node it should forward to based on the rate allocation algorithm, encapsulates it in a UDP packet, and forwards it to the selected node. For a better performance, we recommend that it would be placed as close to the mobile nodes as possible. This would reduce the number of hops required to relay packets.

Switching the DLB service on and off is easy. It can be done by a start/stop of the Agent process and an update to the routing table. One of the reasons to choose IP over UDP is that the firewalls are placed between mobile nodes and the Internet by the cellular operators for security purposes. So we cannot use the techniques like IP over IP or NAT directly because the modified IP packets will be dropped by the operator’s firewall. Our experimentation result shows that the UDP packet encapsulation is fast with a small overhead.

We have conducted several experiments with this tunnelling based DLB system. The results are presented in Section 7.4. We found that it works well for the UDP traffic, but when it comes to the TCP traffic its performance is unstable. One serious problem with this implementation is that our tunnelling based DLB prototype is a network layer system, so it is unaware of the packet ordering. This can significantly degrade the performance of TCP. Packets re-ordering can happen quite often when the traffic is split over multiple different operators. This was one the main motivations to use an alternative solution in our Neptune approach and prototype, which will be described in the next section.
7.3 Neptune Architecture and Prototype

As described previously, the performance of TCP can be significantly degraded due to the arrival of out-of-order packets [18]. While in the DLB system, packet reordering can happen frequently as the long-range links of the cooperative nodes have different rates and delays. In TCP, the out-of-order packets would result in duplicate acknowledgements (DUPACKs) from the receiver. The sender, which receives these DUPACKs, however, cannot differentiate if it is due to an out-of-order delivery or that a packet was lost and DUPACKs are the result of subsequently received packets. Current TCP designs and implementations were optimized for wired networks where out-of-order arrivals are rare and packet loss due to transmission errors is small. So, TCP misinterprets the out-of-order packet delivery as a packet loss due to congestion. These assumptions are no longer true in our DLB wireless and heterogeneous cooperation environment.

A simple and quick fix is to intentionally drop the DUPACKs, but this would be a band-aid solution. It does not solve the fundamental problem, and can cause side effects when dropping useful DUPACKs. Some other solutions [53, 32] use timestamps or additional TCP header bits to detect spurious retransmission and therefore eliminate the retransmission ambiguity. In [73], the authors propose RR-TCP to adaptively vary dupthresh to avoid false fast retransmits proactively. Other solutions such as TCP-DOOR [70], designed for MANET environment with occasional out-of-order packets, detects out-of-order packets by using additional sequence numbers. TCP-PR, proposed in [20], neglects DUPACKs completely, and relies solely on timers to detect packet loss. However, those solutions only consider the case of one single congestion path, which does not correspond to our DLB scenario.
where each link results in an independent transmission path.

According to the specific type of packet reordering that happen in our DLB system, where the packets are a mix of multiple independent transmission paths, we consider a solution that addresses the congestion control for each cooperative link separately. Our prototype Neptune is based on this basic idea.

As a proxy-based approach, Neptune shares a similar structure as the previously introduced tunnelling-based prototype. It also consists of two components: an Agent running in each mobile node and a Remote Proxy which is accessible from the Internet. Unlike the tunnelling based prototype which deals with the packets at the network level, Neptune is working at the transport level, to resolve the packet re-ordering problem. The Neptune system is described below.

Inspired by the Indirect TCP or I-TCP [16], Neptune splits one TCP link into several connected TCP links. A local NAT is added to complete the data collection task. When the application initiates a TCP connection to a remote application server, NAT is applied to redirect the connection to a dummy service stub in the Agent. The application is unaware of this change, and feels as if it is connected to the remote application server. This is quite different from the tunnelling based approach.

For the uplink, once the Agent captures the traffic stream of the application, the cooperation starts. It first transforms the traffic data into its own message format with a sequence number for each message. Notice that those messages are different from the TCP packets received from the application. The message is currently set to a fixed size of 4KB, which defines the minimum data unit to be transferred within Neptune. The scheduler then
dispatches those messages to a group of cooperating nodes according to the rate allocation algorithm. The Agents in other cooperating nodes forward the messages to the Remote Proxy through independent TCP links. As a result, each of these links handles its own congestion independently. Eventually, the received messages from all the cooperating nodes are reassembled at the Remote Proxy according to their sequence numbers. The Remote Proxy then establishes another TCP connection to the destination application server and forward the data to it.

For the downlink, the data sent back from the application server can be obtained from the same TCP link which is connected to the application server. The Remote Proxy then transforms the traffic data into messages with sequence numbers. The scheduler dispatches them back to the cooperating nodes through the same TCP links that are previously connected from them. Each of the cooperating nodes forwards the messages to the source node. The Agent at the source node reassembles and delivers them to the application through the service stub.

This approach uses separate TCP links, which allows independent congestion control, for the links between each cooperating node and the Remote Proxy. From our experimental results (Section 7.4), we show that this approach can effectively avoid the performance fluctuation found in the tunnelling based prototype. As we can see, Neptune splits the original TCP connection from the application to the application server into three separate TCP connections: from the application to the Agent stub, from the Agent to the Remote Proxy, and from the Remote Proxy to the application server.

We implemented the Agent of Neptune on the Google Android platform. It runs as a background service, but it can be controlled through an Android
Activity with UI. A screen shot of Neptune Agent controller is shown in Figure 7.2. Most of the job is done automatically in the background, and the user can simply switch it on and off through a single button. Note that in order to run Neptune, the user needs to have root privileges on the Android phone to gain permission for certain API calls.

![Figure 7.2: Neptune Agent controller on Google Android platform.](image)

![Figure 7.3: A testing tool for DLB prototype.](image)

### 7.4 Experimentation

First, we conducted several experiments with the tunnelling-based prototype. The experimental testbed consists of a computer which serves as the Remote Proxy, and the Agents running on multiple laptops with PC cards. We use two types of cellular PC cards: an HSDPA Sierra Wireless Aircard 860 from AT&T
and a 1xEVDO Rev. A Pantech PX-500 from Spint (Table 7.1). The laptops are running the Fedora Linux with kernel 2.6.21. The Click Modular Router is version 1.6.0. The reasons why we choose to use laptops, rather than a direct implementation on the phones, is that substantial kernel modifications are necessary in order to make the tunnelling based prototype work, and the Click Modular Router is only available on the desktop Linux not on the Linux based mobile phones yet.

<table>
<thead>
<tr>
<th>Manufacture</th>
<th>Model</th>
<th>Technology</th>
<th>Operator</th>
<th>Software OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motorola</td>
<td>Droid</td>
<td>1xEVDO Rev.A</td>
<td>Verizon</td>
<td>Android OS 2.1 Linux Kernel 2.6.29</td>
</tr>
<tr>
<td>HTC</td>
<td>G1</td>
<td>HSPA</td>
<td>T-Mobile</td>
<td>Android OS 1.6 Linux Kernel 2.6.29.6</td>
</tr>
<tr>
<td>Apple</td>
<td>iPhone 3G</td>
<td>HSDPA</td>
<td>AT&amp;T</td>
<td>iPhone OS 3.1.2</td>
</tr>
<tr>
<td>Sierra Wireless</td>
<td>Aircard 860</td>
<td>HSDPA</td>
<td>AT&amp;T</td>
<td>Fedora 7 Linux Kernel 2.6.21</td>
</tr>
<tr>
<td>Pantech</td>
<td>PX-500</td>
<td>1xEVDO Rev.A</td>
<td>Sprint</td>
<td>Fedora 7 Linux Kernel 2.6.21</td>
</tr>
</tbody>
</table>

Table 7.1: DLB Testing Environment Setup

The measurements we take aim at determining the overhead cost and providing a proof of concept for DLB. We develop and run a UDP speed test program that sends packets at a constant rate higher than what is sustainable by each of the cellular PC cards but still acceptable by the Internet. We use a 500KB/s data rate with a UDP packet size of 1KB. Table 7.2 summarizes our measurements. As we can see for the protocol with no ordering requirement such as UDP, our measurements indicate that tunnelling based DLB prototype can be a very effective solution to boost the throughput with very little overhead. However, through our experiments, we also observe that for the TCP protocol, which has strict constraints on the packets ordering, our tunnelling based DLB prototype fails to give consistent results and
the performance fluctuated substantially (sometimes reaching a throughput even lower than what a single path can achieve). From our investigation, out-of-order packets are the source of the performance hit as discussed in Section 7.3.

<table>
<thead>
<tr>
<th>Time</th>
<th>HSDPA</th>
<th>1xEvDO</th>
<th>DLB Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>9 am</td>
<td>920.8</td>
<td>1214.4</td>
<td>2037.6</td>
</tr>
<tr>
<td>12 pm</td>
<td>890.4</td>
<td>1249.6</td>
<td>2046.4</td>
</tr>
<tr>
<td>4 pm</td>
<td>935.2</td>
<td>1370.4</td>
<td>2148.0</td>
</tr>
<tr>
<td>8 pm</td>
<td>934.4</td>
<td>1336.8</td>
<td>2218.4</td>
</tr>
</tbody>
</table>

**Table 7.2:** UDP download speed tests for tunnelling based DLB. (kbps)

We also conducted extensive experiments with our Neptune prototype. Neptune runs on existing smart phones and aims at solving the problems faced in the tunnelling based DLB prototype. Neptune is implemented for the Google Android platform. In our experiments, we use two types of Android phones, the 1xEVDO Rev. A Motorola Droid from Verizon and the HSPA HTC G1 from T-Mobile (Table 7.1). We also include an Apple iPhone 3G from AT&T to cover more cases. Because Neptune is not designed for iOS running on the iPhone 3G, we also use a laptop tethered with the iPhone 3G, so we can use the PC implementation of Neptune. This is a similar method that we used in the experiments of the tunnelling-based prototype.

To measure the performance of Neptune, we also created a testing application (Figure 7.3) to test downloading/uploading with different sizes using HTTP (Most of the mobile traffic is HTTP e.g., web browsing, the Internet radio, YouTube video). In the debug mode, the Neptune Agent can record the transient speed (at the second level), so that we can monitor the details of the running performance for each cooperative node.

As expected, the specific results vary depending on the location and the
7.4. EXPERIMENTATION

time. Most of our experiments are conducted around the Northeastern University campus in Boston. Figure 7.4 and 7.5 show the performance of downloading and uploading of the Neptune DLB system with two Verizon Droid phones. As expected, the Neptune DLB system can almost double the throughput in both downloading and uploading. We also observe that the variance of the single link upload speed is smaller than the single link download speed. This can be explained by the fact that there is less upload traffic in comparison to the download traffic. With less competition, the upload speed graph looks smoother than the download speed graph.

Figure 7.6 and 7.7 show the performance of downloading and uploading of the Neptune DLB system with one Verizon Droid and one T-Mobile G1. From our measurements, the average speed of the Verizon network is around 900 kbps for downloading, and 600 kbps for uploading; T-Mobile network is much slower with an average speed of 500 kbps for downloading and 250 kbps for uploading. Therefore, in the cooperative DLB system, the slow network users can benefit even more.

We also experimented the Neptune DLB system with one Verizon Droid and one AT&T iPhone 3G. Similar results can be found in Figure 7.8 and 7.9. However, the upload throughput of iPhone 3G looks strange. It displays a periodical pattern. This is because the Neptune Agent is not actually running on the iPhone 3G. Instead, as we mentioned it is running on a laptop tethered with iPhone 3G. The OS in the laptop is the desktop Linux. It has a much larger transmission buffer, which causes our transient speed calculation to be inaccurate. Nevertheless, the average speed over a long period is still accurate.

When we bundled two phones from the same operator, we found that oc-
occasionally the aggregated throughput of DLB is not improved by the "usual" level and the performance gain is discounted, as we discovered that both phones are transmitting at lower rates. It is still unclear what is the exact cause of this problem, but we guess one reason can be that if a phone is assigned to a frequency carrier with other phones by the operator, they compete with each other for the available bandwidth. This causes each phone to perform at the lower rate. However, if this is the case, it would not affect the effectiveness of the DLB system in the real world, because even if all cooperative phones are assigned to the same frequency carrier, most likely there are also other mobile phones on that frequency carrier, and with DLB we can still get multiple portions of the available bandwidth through competition.

![Download Test](image1)

**Figure 7.4:** Downloading throughput of Neptune DLB with two Verizon Droid phones in cooperation.

![Upload Test](image2)

**Figure 7.5:** Uploading throughput of Neptune DLB with two Verizon Droid phones in cooperation.
7.4. EXPERIMENTATION

Figure 7.6: Downloading throughput of Neptune DLB with one Verizon Droid and one Tmobile G1 in cooperation.

Figure 7.7: Uploading throughput of Neptune DLB with one Verizon Droid and one Tmobile G1 in cooperation.

Figure 7.8: Downloading throughput of Neptune DLB with one Verizon Droid and one AT&T iPhone 3G in cooperation.

Figure 7.9: Uploading throughput of Neptune DLB with one Verizon Droid and one AT&T iPhone 3G in cooperation.
CHAPTER 8

Conclusion

In this thesis, we investigate the power of distributed cooperation in networks of mobile nodes equipped with two wireless interfaces: (1) a long-range bandwidth-limited, and (2) a short-range high-bandwidth. We explore cooperation at multiple levels of the network stack and across the local ad hoc network and the cellular network.

The fundamental motivation for cooperation is to exploit diversity. In wireless mobile networks, diversity occurs in several forms. At the physical layer, in urban environments, the link quality can significantly vary over short periods of time (due to mobility and multipath fading) and also over longer periods of time (due to shadowing). We exploit the channel diversity by combining the best links to a base station, surrounding a specific node, and build an efficient combined link. At the link layer and above, diversity occurs in terms of independence of traffic characteristics. We exploit traffic diversity by bonding multiple links together, multiplexing the traffic across users’ connections and operators’ coverage.

We introduce a framework for exploiting distributed diversity in hetero-
CHAPTER 8. CONCLUSION

geneous wireless networks through cooperation. We propose several cooperation protocols and architectures, and analyse their performance. The performance of the proposed mechanisms is derived analytically, or through simulations and compared to experimental data whenever possible. The experimentation testbed includes both software-defined radios and mobile phone implementations. We show that our physical layer signal combining mechanism can theoretically lead to operating a receiver at orders of magnitude lower signal strength than traditionally. Furthermore, our experiments demonstrate significant improvement in performance can be reachable, despite the limitations of the testbed. At the link and transport layers, we show that the perceived throughput can scale linearly with the number of cooperating nodes. The proposed mechanisms are implemented on the Android platform and evaluated on real networks both within a single cellular operator and across operators.

We demonstrated that distributed diversity through cooperation has a huge potential to improve the performance of wireless networks. While some of our proposed mechanisms can readily be used, many are just a first step towards leveraging cooperation. A major open question is how to efficiently enforce cooperation preventing malicious users. Another direction of research is how to expand cooperation beyond a single-hop short-range network and how to route, multiplex, and schedule cooperation traffic.
Bibliography


[9] **iPhone vs. Pre: Satisfaction bakeoff.**

[10] **Mobile broadband still crawling at below 1Mb, despite ‘up to’ 7.2Mb claims.**
http://mobile.broadbandgenie.co.uk/broadband-news/mobile-broadband-still-crawling-at-below-1mb-despite-up-to-7mb-claims.

[11] **Motorola’s mesh networks.**


