Wireless Networks: New Models and Results

Thesis by

Radhika Gowaikar

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

California Institute of Technology

Pasadena, California

2007

(Defended July 25, 2006)

Acknowledgements

When I now think of the person who came to Caltech in 2001, seeking a Ph.D., it feels like she was making an enormous leap of faith. The leap was made almost casually, without too much thought, but the fact that I am all in one piece five years later owes itself to the many people who ensured that I landed on my feet.

Primary among them is my adviser, Babak Hassibi. His acute intuition and his vision of what constitutes an important problem have made this thesis possible. Babak's enjoyment of and dexterity with complicated mathematical expressions is unmatched. He has been available for discussion at all times of night and day (literally!) and it has been my privilege and my pleasure to have worked with him.

A large portion of this thesis was accomplished during a summer spent at Bell Laboratories, with Bert Hochwald as my mentor. Bert has an exceptional ability to reach down to the hairy details of a problem and draw a conclusion about the big picture. I have learned a lot from him.

Sincere thanks go to Profs. Michelle Effros, Tracey Ho, Bob McEliece, Leonard Schulman, and P. P. Vaidyanathan, for being on my candidacy and thesis committees and providing valuable feedback. I am grateful to Shirley Beatty for her pleasant and efficient help over the last five years.

Amir, Cédric, Chaitanya, Haris, Masoud, Mihailo, and Yindi have demonstrated that being at work eight hours a day can actually be a lot of fun. They made the lab a place where I felt at home and where I could simply hang out. (In addition, this also meant that I did not feel guilty about occasionally slacking off at work. It was a win-win situation.) I thank these people for indulging me in many ways and humoring me through all the long, wonderful and pointless conversations, not least

those involving the meaning of life. It is difficult to express in words how much their company and friendship have meant to me. I can only hope that I gave as much as I got.

I am fortunate to have enjoyed a life outside of research during my stay at Caltech. Anelia, Anu, Peter, and Sujata have been the people I have called on to share my moments of crisis and my moments of joy. They have given me some of my most treasured memories and made it very difficult to leave Pasadena. Greg and the Caltech Y, Desiree and the Women's Glee Club, Peter and the Tai Chi Club, and Jim and the ISP have enriched my life beyond measure. In these and a thousand other intangible ways, Caltech has broadened my horizons. It is to this Caltech Experience that this thesis is dedicated.

Finally, I acknowledge my parents for being there so unfailingly. Their patience and support, especially in my more wayward phases, have meant the world to me. For allowing me to make my leaps of faith, I shall always be grateful.

Abstract

Wireless communications have gained much currency in the last few decades. In this thesis we present results regarding several wireless communication systems, in particular, wireless networks.

For some time now, it has been known that in an ad hoc network in which nodes share the wireless medium, and the connection strengths between nodes follow a distance-based decay law, the throughput scales like $O(\sqrt{n})$, where n is the number of nodes. In Chapter 2 we introduce randomness in the connection strengths and examine the effects of this on the throughput. We assume that all the channels are drawn independently from a common distribution and are not governed by a distance-decay law. It turns out that the aggregate information flow depends strongly on the distribution from which the channel strengths are drawn. For certain distributions, a throughput of $\frac{n}{(\log n)^d}$ with $d > 0$ is possible, which is a significant improvement over the $O(\sqrt{n})$ results known previously. In Chapter 3, we generalize the network model to two-scale networks. This model incorporates the distance-decay law for nodes that are separated by large distances, while maintaining randomness in close neighborhoods of a node. For certain networks, we show that a throughput of the form $n^{\frac{1}{t-1}}/\log^2 n$ is achievable, where $t > 2$ is a parameter that depends on the distribution of the connection at the local scale and is independent of the decay law that operates at a global scale.

In Chapter 4, we consider a model of an erasure wireless network, in which every node is connected to certain other nodes by erasure links, on which packets or bits are lost with some probability and received accurately otherwise. Each node is constrained to send the same message on all outgoing channels, thus incorporating the

broadcast feature, and we assume that there is no interference in the network, other than through the possible correlation of erasure occurrences. For such networks and in certain multicast scenarios, we obtain the precise capacity region. This region has a nice max-flow, min-cut interpretation and can be achieved using linear codes. We do require the side-information regarding erasure locations on all links to be available to the destinations. Thus, we have the capacity region for a non-trivial class of wireless networks.

Recent results for wireline networks show that in several scenarios, it is optimal to operate these networks by making each link error-free. In Chapter 5, we consider Gaussian networks with broadcast and interference, and erasure networks with broadcast, and show that in the presence of these wireless features, it is suboptimal to make each link or sub-network error-free. We then consider these networks with the constraint that each node is permitted to either retransmit the received information or decode it and retransmit the original source information. We propose a greedy algorithm that determines the optimal operation for each node, such that the rate achievable at the destination is maximized. Further, we present decentralized implementations of this algorithm that allow each node to determine for itself the optimal operation that it needs to perform.

In Chapter 6, we consider a point-to-point communication system, involving multiple antennas at the transmitter and the receiver. These systems can give high data rates provided we can perform optimum, or maximum-likelihood, decoding of the received message. This problem typically reduces to that of finding the lattice point closest to a given point x in N-dimensional space. This is an integer least-squares problem and is NP-complete. The sphere decoder is an algorithm that performs decoding in an efficient manner by searching for the closest point only within a spherical region around x. In Chapter 6, we propose an algorithm that performs decoding in a sub-optimal manner by pruning the search region based on the statistics of the problem. This algorithm offers significant computational savings relative to the sphere decoder and allows us to tradeoff performance with computational complexity. Bounds on the error performance as well the complexity are presented.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The problem of communicating information has been approached in many novel and creative ways in the past few centuries. Before the 1800s, methods such as fires, drums, mirrors and carrier pigeons were commonly used to send messages. In the 1800s, our understanding of electromagnetic phenomena advanced rapidly and made inventions such as the telephone and telegraph possible. When Marconi sent his radio signal across the Atlantic in 1901, wireless communication first became a viable approach to the problem of sending data from one point to another.

In the past century, and especially in the past forty years, we have seen many advances in the area of communication. Today, communication systems come in many, and increasingly sophisticated, flavors. Systems can range from one sender and one receiver using walkie-talkies, to millions of users interacting with each other through a network as in the internet, the telephone system or the cellular system. Thanks to the wide range of sizes, functionalities, precise models and performance measures of interest, a rich field of research has come to be associated with these systems.

Many communication systems that are deployed in the world today are either entirely or predominantly wireless. In the future, we expect the wireless phenomenon to continue to flourish as these systems become ubiquitous. Also, we have begun to expect data such as video and multimedia to be transmitted over these systems, rather than just voice. This means that the bit-rate and reliability of the transmission must increase as we go into the future.

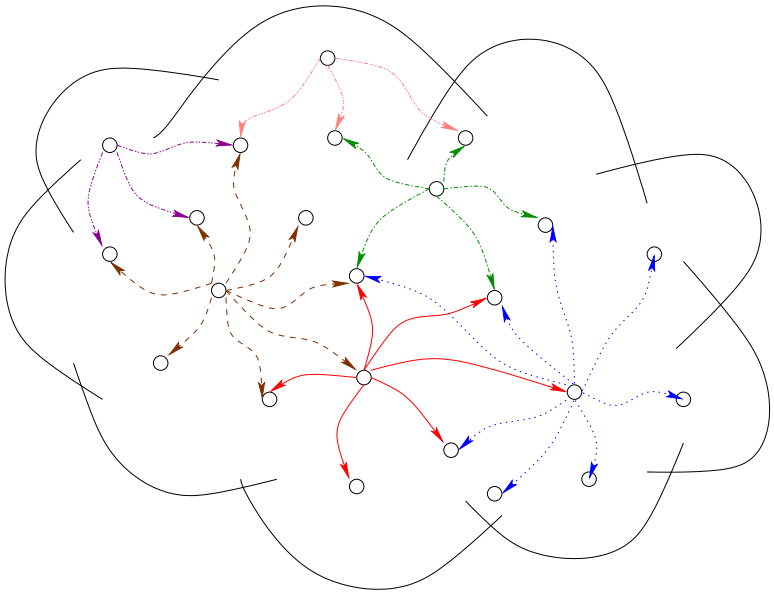


Figure 1.1: Depiction of a wireless network. It is a shared medium with broadcast and interference. The strength of the links fluctuates with time.

1.1 Features of Wireless Systems

In order to efficiently design and operate the communication systems of today and tomorrow, we first need to characterize and analyze them in sufficient depth. The features associated with the wireless medium need to be understood and exploited. One prominent feature of the wireless medium is that it is shared by many users and there is no allocation of resources enforced by the medium to begin with. Properties of broadcast and interference are also important. The former implies that when any user transmits a message, many other users are likely to hear it. On the flip side, when two or more users are simultaneously transmitting to their respective destinations, the destinations hear the messages not intended for them in addition to those intended for them, leading to interference. Finally, the most unique feature of the wireless connection is that its strength varies over time. The connection between two nodes can be very strong at some times and very weak at other times. This phenomenon, called fading, is largely because of the random fluctuations in the medium and is enhanced by factors like the mobility of users. In addition, since there may be several paths for the message to reach the destination, each with different delays, multiple

copies of the message can reach the destination at different times, leading to the multipath effect.

Thus, the wireless connection has an inherently probabilistic nature. This makes the connections unreliable, but recently developed techniques turn this into an advantage using the concept of "diversity." Put very simply, this is done by transmitting messages over several links simultaneously. Since the probability of *all* the links being simultaneously weak is very small, this increases the reliability of the system. Thus, one can view the randomness introduced by fading as an advantage. Other advantages of wireless systems are that they require less investment in infrastructure and allow mobile systems to be connected temporarily.

1.2 Some Important Issues

There are many issues that merit attention in the analysis of a communication system. The rate at which bits get across from a sender to a receiver is an important issue. This can be characterized by the throughput or the capacity of the network. Another important feature is the error performance or the rate-distortion behavior of the system. Systems in which performance degrades gracefully as the channel conditions deteriorate are preferred over systems that are very sensitive to the quality of the channel.

Other practical issues that are of interest involve delay or the time between the transmission and reception of the message. Systems in the real world usually have strict constraints on the delay that can be tolerated. These systems also have to be robust to the failures of certain links or nodes. In addition, we may require the different nodes in a system to be able to make their own decisions and operate in a decentralized manner.

Thus, there are many competing requirements that have to be met in a well-designed system. For the purposes of analysis, however, it often becomes necessary to find a suitable model of the system that enables us to focus on a small number of issues. At the same time, in order for the model to be relevant, it has to be close to

reality. Thus, modeling and characterizing a system is the first challenge in analyzing it.

A large part of this thesis involves proposing new models for wireless networks that mirror real networks closely, yet remain tractable. For these models, several issues that are important from a theoretical as well as practical point of view are investigated. These include capacity, decentralized operation, scheduling and throughput. Results of a problem involving point-to-point systems are also presented. In these, a method of reducing the receiver complexity of space-time systems is proposed. An understanding of various probabilistic tools as well as mathematical concepts, such as random graphs, matrix analysis and Markov chains has facilitated many of these results.

1.3 Problems and Contributions

What follows next is a brief summary of the various questions addressed in this thesis and the contributions made towards answering them.

1.3.1 Wireless Networks with Random Connections

Current multiple user wireless systems (e.g. cellphone networks) typically employ some centralized infrastructure (e.g. basestations) in order to operate. Ad hoc networks deal with the issue of facilitating communication between pairs of nodes in the absence of such infrastructure. The work of Kumar and Gupta in 2000 [35] has provided great insight into the behavior of these networks.

They considered a network with n nodes distributed over a certain area, where the channel strength between any two nodes was inversely proportional to some power of the distance between them. With this model, as n increased, only about $\sqrt{n}$ messages could be simultaneously transmitted across randomly chosen transmitter-receiver pairs. This meant that the per user throughput actually decreased as $\frac{\sqrt{n}}{n} = \frac{1}{\sqrt{n}}$, indicating that large ad hoc networks were not practically feasible. Since then, most other work based on similar network models has only reinforced this conclusion.

The question we ask in Chapter 2 is whether these results can be improved in any way using a different network model. We propose a model in which the geographical location of nodes does not play a role in determining connection strength. Instead, connection strengths are independently and identically distributed (i.i.d.) according to a particular probability distribution function (pdf). Recent research on the connectivity of ad hoc networks supports such a model. This model is also suitable for indoor networks, where obstructions and reflections cause scattering which dominates the line-of-sight component. This random model is amenable to analysis and can improve throughput to $n/\log^c n$ for some constant c [51, 54].

We develop a general approach that works for any pdf that the connections may be drawn from. For example, if the pdf is a simple Bernoulli, we find that the value of the Bernoulli parameter that maximizes throughput is $\frac{\log n}{n}$, which is surprisingly low. This means that each node has only $\log n$ neighbors out of a potential $n-1$; in fact, this is just enough to ensure connectivity of the network. At this connectivity the throughput scaling is quite encouraging. It increases as $n/\log^2 n$, which is only slightly sublinear.

In order to compare performance with the Kumar and Gupta model, we consider the distribution of received signal powers when a single node transmits in their network setting. Assuming that the i.i.d. connections in our random model are drawn from this distribution, we show that throughputs of the form $n/\log^c n$ are achievable, for constant c. This is significantly better than the $\sqrt{n}$ obtained in their deterministic model and tells us that randomness in the connections offers great advantages. The maximum benefit comes from the fact that only around $\log n$ hops are required in our model, as against the $\sqrt{n}$ required in theirs.

In reality, the distance-decay effects that are represented in the Kumar-Gupta model are far field effects and only set in over long distances. On the other hand, the i.i.d. connections are expected to model reality over shorter distance, where the far field effects are not set in. Thus, though a network with i.i.d. connection strengths can behave very differently from one with entirely deterministic connection strengths, a realistic network model should incorporate both random and location-

dependent features. A possible model for such a network is one in which nodes within a certain radius enjoy i.i.d. connections with each other, but face a distance-decay law with nodes outside that radius. Such a model is proposed in Chapter 3. For this model, we use a combination of the techniques used by Kumar and Gupta and those developed in Chapter 2 in order to find suitable schedules of relays. More specifically, we develop a skeleton of a schedule using the ideas of tessellations proposed in [35] and then fill up the rest of the schedule using ideas from the purely random model. We are able to obtain a result regarding the achievable throughput for a general combination of the distance-decay law and the pdf for the random connections. For example, we show that an aggregate throughput of the form $n^{\frac{1}{t-1}}/\log^2 n$ is achievable, where $t > 2$ is a parameter that depends on the distribution of the connection at the local scale and is independent of the decay law that operates at a global scale. For $t < 3$, this offers a large improvement over the $O(\sqrt{n})$ results that are known for the purely distance-based models.

1.3.2 Wireless Erasure Broadcast Networks

The results for the ad hoc networks of Chapter 2 are in the form of scaling laws and hold for asymptotically large networks. For smaller networks we are interested in characterizing the capacity exactly. This problem has proved to be quite challenging – no complete solution is known even for the simple relay channel which consists of three nodes. Outer bounds on the capacity region are known and are usually obtained using cut-capacities [71]. To obtain these bounds the nodes of the network are divided into two parts. Assuming that nodes within each part can cooperate fully we obtain the mutual information between the two parts. Until recently, no non-trivial networks were known in which these bounds were achieved.

Thus, we are faced with the problem of finding a model that is realistic yet tractable. We consider a packet-based network modeled by a directed acyclic graph with several independent information sources that are desired by several destinations. All nodes can act as relays. Each edge erases packets with some probability. The

network is wireless in that the same message has to be broadcast on all outgoing edges of a node. However, we assume that some multiple access scheme is in place that eliminates interference among incoming edges.

For general channels and in the absence of the broadcast property this forms the wireline multicast problem. This was solved in 2000 by Ahlswede et al. [68] whence the field of network coding took off. They showed that the cut-capacity outer bounds were achievable and that it was optimal to operate each edge at or below its capacity in an error-free manner.

From our other work (described in the next section) we know that wireless networks differ significantly from wireline networks in that it is sub-optimal to make each link or sub-network operate without error. Therefore, we do not attempt to make each erasure channel error-free. Instead, nodes use random encoding functions that are known to the destinations. In addition, we assume that the erasure locations are also known to the destinations as side-information. We show that with this setup, the cut-capacity outer bounds can indeed be reached (Chapter 4).

Due the available side-information, the destinations are in a position to simulate the network for every possible combination of messages that the sources might be transmitting. It turns out that, with probability going to one, only one combination of messages can produce the observed output, leading to successful decoding. Thus we have a capacity region for wireless erasure broadcast networks [1, 63]. Note that this result is valid for any size and topology of the network.

We have also generalized this result. We show that restricting ourselves to *linear* random encoding functions is sufficient for achieving capacity. Capacity results for correlated erasures are also obtained. Changing the packet erasure channels to erasure channels with any discrete input alphabet (e.g., binary) leads to similar results. However, for packet erasure channels, the overhead of providing side-information regarding erasure locations is small, especially for long packets. More interesting results for broadcasting over our network model have also been obtained [2].

1.3.3 Optimal Policies for Decode/Forward Networks

The network problems described above consider throughput and capacity and are of a fundamental theoretical nature. In Chapter 5 we look at a practical problem of concern in real networks. While capacity regions provide us with the absolute bounds on how much information a network can support, ways of reaching those bounds are often very expensive in terms of time, computation and memory. Also, a simple and cheap node that is part of a large network may be incapable of performing the required operations. Therefore, we consider wireless networks with only one source and one destination in which other nodes act as relays and are restricted to performing one of only two operations. These are retransmitting the received information or decoding it and transmitting the original codeword. We find the optimal operation for each node such that the rate from the source to the destination is maximized.

Had this been a wireline network, the optimal operation for each node would be to decode, and each link would operate error-free. However, we show through examples that for a network with the broadcast property, it is sub-optimal to make links or sub-networks operate error-free [20, 62].

The networks we consider are of two types. One is a Gaussian network, modeled by a directed acyclic graph, where there are channel coefficients associated with each edge. Broadcast and interference act in the usual way. Each node has a power constraint and experiences additive Gaussian noise. To forward, a node scales the received message appropriately and retransmits. The second network differs from the wireless broadcast erasure network described in the previous section in one way. In order to be able to forward, we assume that links can take erasures as inputs and that these are received as erasures. In both networks we ask a node to decode only if it can do so without error. That is, if it can support the rate that the source is trying to deliver to the destination. Thus, asking a node to decode puts a constraint on the rate.

If there are V nodes in the network and each relay node is permitted one of two operations, there are 2^{V-2} policies possible. We present a greedy algorithm that goes

over at most $V-2$ policies and returns the optimal one, thus avoiding an exponential search [19]. We have also developed decentralized implementations of this algorithm that converge to the maximum rate iteratively. These require one bit of feedback at every iteration. Furthermore, the algorithm can apply to a wider class of networks than the two mentioned here.

1.3.4 Efficient Near-ML Decoding via Statistical Pruning

While multiple antenna systems promise high rates [28, 37], reliable decoding is important to fully realize this benefit. Unfortunately, decoding in these systems usually involves high computational complexity and is currently recognized as the major bottleneck in the design and use of space-time systems. Though there exist sub-optimal decoding algorithms that run in cubic time, the error performance of these is significantly worse than that of the optimum decoder. Therefore, finding efficient and accurate decoding algorithms is important.

For a typical multiple antenna system, the optimum, or maximum-likelihood (ML), decoding problem that the receiver has to solve is a minimization over the discrete signal space, the size of which is exponential in the problem dimension. In fact, it has the form of a standard integer least-squares problem, which is known to be NP-hard. This often means that the best way of performing the required minimization is by exhaustively searching over the discrete signal space.

The sphere decoder is an algorithm that tries to avoid the exhaustive search by restricting the search to points within a specific sphere that is certain to contain the minimizer. While the sphere decoder decreases complexity to a reasonable extent, further reduction, especially in regimes of low SNR and high problem dimension is desirable.

In Chapter 6, we propose a decoder that increases efficiency by restricting the search to a non-spherical region that is much smaller than the sphere of the original algorithm. Though this reduces the search space, there is a price to pay in terms of performance. Because of the asymmetry of the new search region there is a small

probability that the decoder output will not be the ML-output, thus making the decoder sub-optimal.

However, the new search region is statistically pruned in a careful manner keeping in mind the stochastic nature of the channel and the noise. Therefore, with high probability, we manage to simultaneously get the benefits of low complexity and high performance. We are able to quantify and control the sub-optimality and can often design a search region that operates at a desired trade-off between complexity and performance [97, 96, 98]. Our schemes can reduce complexity (with respect to a state-of-the-art sphere decoder) by a factor of 240 for a 50 antenna system with 4-QAM. With fewer antennas we expect smaller gains – a factor of 7 reduction for a 12-antenna system with 64-QAM. This is achieved while keeping performance within 0.1 dB of the optimal.

In the presence of coding, the size of the problem depends on the exact coding scheme as well as the number of antennas. This number of virtual antennas is expected to be much larger than the number of actual antennas. Thus, the decoding problem has to be solved in a high dimensional setting. The real benefits of the pruned decoder materialize in this regime.

Each of the problems considered in this thesis raises interesting issues that can be investigated further. We present discussions, summary and directions for future work in Chapter 7.

Chapter 2

Achievability Results for Random Wireless Networks

The model of wireless ad hoc networks in which connection strengths are based on the distances between nodes is well-studied. In this chapter, we motivate and propose a substantially different model, in which channel connections are entirely random. We assume that, rather than being governed by geometry and a decay-versus-distance law, the strengths of the connections between nodes are drawn independently from a common distribution. We show that the throughput behavior, as a function of the number of nodes n, is strongly dependent on the channel distribution. For certain distributions, a throughput of $\frac{n}{(\log n)^d}$ for some $d > 0$ is achievable, which is significantly greater than the $O(\sqrt{n})$ results obtained for many geometric models.

2.1 Introduction

An early study of traffic flow in shared-medium wireless networks appears in the seminal work of Gupta and Kumar [35]. They show that in a grid network of n nodes on the plane having a deterministic power-scaling law, $O(\sqrt{n})$ transmitters can talk simultaneously to randomly chosen receivers. Similar results for networks with randomly placed nodes can also be obtained (see, for example, [34] for a recent account). Different models can yield somewhat different conclusions [25, 27, 29, 33, 36, 38, 39, 40, 41]; nevertheless, if we do not permit the transmitter/receiver pairs to approach one another [30], or for very low attenuation laws [39], the model of a

power decay law (as a function of distance) seems to yield a network in which the number of nodes that can talk simultaneously grows much slower than n. Network models that incorporate channel fading as well as geometric path loss have also been proposed [47, 46] but the scaling behavior of these is not much different from that of [35]. We wish to study networks with a different connectivity model.

The $O(\sqrt{n})$ result in [35] has the following heuristic explanation. If a node wishes to transmit directly to a randomly chosen node (whose distance is approximately $O(\sqrt{n})$ away on average), it has two choices: talk directly, or talk through a series of hops. If it tries to talk directly, the transmitter generates energy in a circle of radius $O(\sqrt{n})$ around itself. However, this energy, which is seen by the intended receiver becomes interference for the $O(n)$ other nodes in the circle. Thus, some constant fraction of the entire network of n nodes is bathed in interference; an undesirable consequence. If it decides instead to talk through hops, the transmitting node can pass its message to a neighbor, who in turn passes it to a neighbor and so on for $O(\sqrt{n})$ hops to the intended receiver. This strategy limits interference to immediate neighbors but ties up $O(\sqrt{n})$ nodes in the hopping process. Although this turns out to be the best strategy, only $O(\sqrt{n})$ simultaneous messages can be passed before all n nodes in the network are involved.

We change the model of the wireless medium from a model based on distance to one based on randomness. In multi-antenna links, a linear increase in capacity (in the minimum of the number of transmit/receive antennas) is obtained when the channel coefficients between the transmit and receive antennas are independent Rayleigh-distributed random variables [28, 37]. It is therefore now generally believed that a rich scattering environment, once thought to be detrimental to point-to-point wireless communications, may actually be beneficial. We show that a similar effect may hold for the expected aggregate data traffic in a wireless network; certain forms of randomness can be helpful.

There are several reasons why one may choose a random model over one that is based on distance. While distance effects on signal strength are important for nodes that are very near or far from each other, many networks are designed with

minimum and maximum distances in mind. Decay laws of the form $1/r^m$ for a fixed $m > 0$ may not be relevant for networks of small physical size. Additionally, through the use of automatic gain control, a radio often artificially mitigates distance effects unless the node is saturated (too close) or "dropped out" (too far). Many first-order signal-strength effects in such networks are then due to random fluctuations in the medium, such as Rayleigh and shadow fading. A distance-power model cannot readily account for shadow-fading since signal strength at the receiver is determined more by the presence of an obstacle blocking the path to the transmitter than by distance. In addition, recent investigations show that the connectivity of ad hoc networks with channel randomness, such as that caused by shadow-fading, is similar to the connectivity in a random graph [48]. Some models that consider channel randomness are studied in [49, 50], where it is shown that the resulting random network has some realistic connectivity properties lacking in a purely deterministic model. We are concerned not just with connectivity but also throughput.

We adopt the premise that randomness can have a first-order effect on the behavior of a network. We assume that the channels between nodes are drawn independently from an identical distribution. We allow the distribution of the channel between nodes to be arbitrary and allow it to vary with the number of nodes n. Our model covers environments where the signal strength at a receiving node is governed primarily by a random event (such as the existence of an obstacle). We believe that the study of such wireless networks with random connections is important for three reasons: First, many real wireless networks have a substantial and dominant random component; second, we show that such networks may have qualitatively different traffic scaling laws than the scaling obtained in geometric models; finally, our results give insight into the connectivity that a network should have to allow large aggregate traffic flows.

In general, any realistic model of a large network should have a model of connectivity that has a balance of randomness and distance-based effects. In [52] one such model is proposed and its throughput is analyzed. Also, [32] uses a "radio model" to show that in the presence of obstructions and irregularities, channels become approximately uncorrelated with one another, and the probability of good links between

nodes that are far apart increases in wireless local area networks (WLANs). The radio model in [32] essentially uses the same independence assumption that we do, but uses distance to determine the probability of a connection link. We show in Section 2.8.1.1 how to apply our traffic-flow conclusions to this radio model to determine a favorable distance between nodes.

2.1.1 Approach

We suppose that the connection strengths between the n nodes of the wireless network are drawn i.i.d. from a given arbitrary distribution. In geometric networks such as [35] a node may communicate its message in hops to nearby neighbors so that it ultimately reaches the intended destination. In our random model, although there is no geometric notion of a near neighbor, we can find an equivalent of a near neighbor by introducing the notion of "good paths," where connections stronger than a chosen threshold β are called *good*. Transmissions to relays and destinations occur along only good paths. By figuratively drawing a graph whose vertices are all the nodes in the network, yet whose edges are only the good paths, we obtain a specific random graph model called $\mathcal{G}(n,p)$, where an edge between any pair of the n nodes exists with probability p. (In our case, p is simply the probability that the connection strength exceeds β_n.) $\mathcal{G}(n,p)$ is a very well-studied object and we leverage some of its known properties to establish node-disjoint routes between sources and their intended destinations. However since we are analyzing a wireless network, we must also account for the effects of interference between all nodes, including those that do not have good connections between them. Fortunately, our use of the goodness threshold β also makes the analysis of message-failures (due to interference and/or noise) tractable. Our analysis yields an achievable aggregate throughput which is a function of the chosen threshold β. A judicious choice of β can maximize this achievable throughput. To complement our achievability results, we also present some upper bounds on aggregate throughput that show that our results are sometimes tight.

2.2 Model of Transmitted and Received Signals

We assume that the wireless network has narrowband flat-fading connections whose powers are i.i.d. according to an arbitrary distribution $f(\cdot)$. Thus, if $h_{i,j}$ is the connection between nodes i and j, then $\gamma_{i,j} = |h_{i,j}|^2$ are i.i.d. random variables with marginal distribution $f(\gamma_{i,j})$. For maximal generality, we allow $f(\gamma) = f_n(\gamma)$ to be a function of the number of nodes n. As an example, consider

$$f(\gamma) = (1-p) \cdot \delta(\gamma) + p \cdot \delta(\gamma - 1) \tag{2.1}$$

where $\delta(\cdot)$ is the Dirac delta-function. This distribution is a simple model of a shadow-fading environment where, for any pair of nodes, with probability p there exists a good connection between them (fading causes no loss), and with probability $1-p$ there exists an obstruction (fading causes a complete loss). In a general network of n nodes, we may let $p = p_n$ be a function of n to represent changes in the geography or network topology as the network increases in size. Although $\gamma = 0$ and $\gamma = 1$ are the only possibilities in the distribution (2.1), we may also introduce values of γ that depend on n. Figure 2.1 pictorially displays an example of wireless terminals whose connections may obey the model (2.1).

The behavior of such a network varies dramatically with p. At the extreme of $p = 1$ no paths are ever blocked and all nodes are fully connected to each other. While this situation permits any node to readily talk to any other node in a single hop, the overall network throughput is low because talking pairs generate an enormous amount of interference for the remaining nodes. If many nodes try to talk simultaneously, the overall interference is overwhelming. At the other extreme of $p = 0$, everyone is in a deep fade; now interference is minimal. However, no nodes can talk at all (we assume a transmission power limit). Thus we have competing effects as a function of p: Increasing p benefits the network by improving connectivity thus allowing for shorter hops, but hurts the network by increasing interference to other receivers. We are led to ask: What p is optimal? What is the resulting network aggregate traffic?

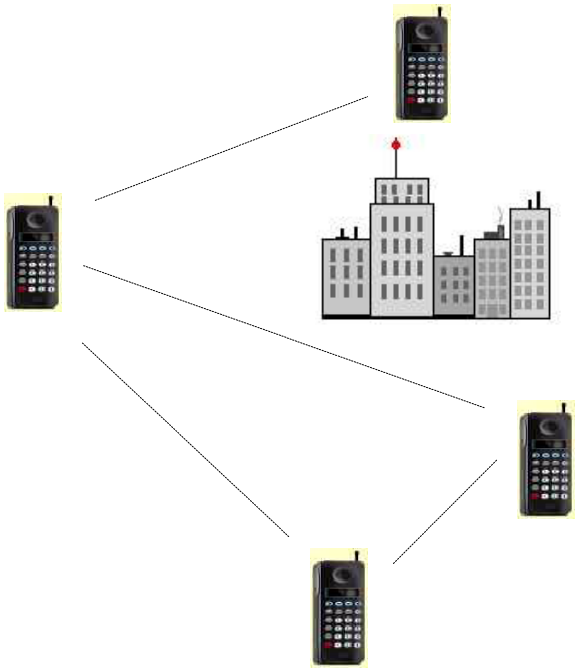


Figure 2.1: Nodes are able to establish connections with each other if there is no object in their path. Equation (2.1) models the presence of an object as a random event where each path has a connection of strength one with probability p, and otherwise has a connection of strength zero.

Is this optimal p likely to be something we encounter naturally? If not, can we induce it artificially? We answer some of these questions but, more generally, we look at how an arbitrary $f_n(\gamma)$ affects the traffic.

2.2.1 Detailed Model

Let the network have n nodes labeled $1, \ldots, n$. Every pair of nodes $\{i, j\}$ $(i \neq j)$ is connected by a channel that is denoted by the random variable $h_{i,j} = h_{j,i}$; there are $\binom{n}{2}$ channel random variables. The channel strengths, $\gamma_{i,j} = |h_{i,j}|^2$ are drawn i.i.d. according to the pdf $f_n(\gamma)$. Once drawn, these channel variables do not change with time.

Node i wishes to transmit signal x_i. We assume that x_i is a complex Gaussian random process with zero mean and unit variance. Each node is permitted a maximum power of P watts.

We incorporate interference and additive noise in our model as follows: Assume

that k nodes $i_1, i_2, \ldots, i_k$ are simultaneously transmitting signals $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$ respectively. Then the signal received by node $j(\neq i_1, \ldots, i_k)$ is given by

$$y_j = \sum_{t=1}^{k} \sqrt{P} h_{i_t,j} x_{i_t} + w_j \tag{2.2}$$

where w_j represents additive noise. The additive noise variables $w_1, \ldots, w_n$ are i.i.d., drawn from a complex Gaussian distribution of zero mean and variance σ^2 ($w_j \sim \mathcal{CN}(0, \sigma^2)$). The noise is statistically independent of x_i.

2.2.2 Successful Communication

In equation (2.2), suppose that only node i_1 wishes to communicate with node j and the signals $x_{i_2}, \ldots, x_{i_k}$ are interference. Then the signal-to-interference-plus-noise ratio (SINR) for node j is given by

$$\rho_j = \frac{P\gamma_{i_1,j}}{\sigma^2 + P\sum_{l=2}^{k} \gamma_{i_l,j}}.$$

We assume that transmission is successful when the SINR exceeds some threshold ρ_0. If the SINR is less than ρ_0 we say that transmission is not possible. Thus, even though $\rho_j \geq \rho_0$, we use $\log(1+\rho_0)$ as the transmission rate. Using $\log(1+\rho_0)$ as the rate, rather than the more precise $\log(1+\rho_j)$, simplifies our analysis.

2.3 Network Operation and Objective

We suppose that k nodes, denoted by $s_1, \ldots s_k$, are randomly chosen as sources. For every s_i, a destination node d_i is chosen at random, thus making k source-destination pairs. We assume that these $2k$ nodes are all distinct and therefore $k \leq n/2$. Source s_i wishes to transmit message M_i to destination d_i and has encoded it as signal x_i. We wish to see how many source-destination pairs may communicate simultaneously. The sources may talk directly to the destination nodes or may decide to communicate in hops through a series of relay nodes.

2.3.1 Communicating with Hops

In general, we suppose that the source-destination pair (s_i, d_i) communicates using a sequence of relay nodes $r_{i,1}, r_{i,2}, \ldots, r_{i,h-1}$. ($h = 1, 2, \ldots$ represents the number of hops.) Define $r_{i,0} = s_i$ and $r_{i,h} = d_i$. The path from s_i to d_i is then $r_{i,0} = s_i, r_{i,1}, r_{i,2}, \ldots, r_{i,h-1}, r_{i,h} = d_i$. In time slot $t+1$ we have nodes $r_{1,t}, r_{2,t}, \ldots, r_{k,t}$ transmitting simultaneously to nodes $r_{1,t+1}, r_{2,t+1}, \ldots, r_{k,t+1}$ respectively. We have nodes $r_{1,t+1}, r_{2,t+1}, \ldots, r_{k,t+1}$ decode their respective signals $x_1, x_2, \ldots, x_k$ and transmit them to the next set of relay nodes in the $(t+2)$th time slot, and so on. A natural condition to impose is that the relay nodes that are receiving (or transmitting) messages in any time slot be distinct so that the messages do not collide. In addition, impose the constraint that relay nodes cannot receive and transmit at the same time. In the rest of the chapter, we refer to these conditions together as the *no collisions* property. In general, we allow $r_{i,t} = r_{i,t+1}$ for any i. This means that a relay can effectively hold on to a message in a time slot; hence h effectively represents the maximum number of hops needed for all the source-destination pairs.

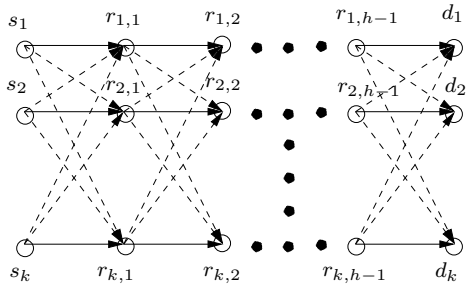


Figure 2.2: Schedule of relay nodes: Source s_i communicates with destination d_i using relays $r_{i,1}, \ldots, r_{i,h-1}$. The solid lines indicate intended transmissions and the dashed lines indicate potential interference. A schedule is valid if it meets the no-collision conditions that a node can receive or transmit at most one message in any time slot and that no node can transmit and receive simultaneously.

2.3.2 Throughput

With the above procedure, we have k simultaneous communications occurring in h time slots. Message M_i reaches the intended destination d_i successfully if it can be

decoded by each relay $r_{i,t}$. Assume that a fraction $1-\epsilon$ of messages reach their intended destinations in this way. Then we define the throughput as

$$T = (1-\epsilon)\frac{k}{h}\log(1+\rho_0), \tag{2.3}$$

where ρ_0 is the SINR threshold, and we are using the natural logarithm. Thus, $\log(1+\rho_0)$ is the sustainable throughput per user if the users do not collide. We multiply this factor by the number of non-colliding source-destination pairs k, divide by the number of hops, and subtract the fraction of dropped messages ϵ. The resulting throughput T depends on n and we sometimes add subscripts to the variables involved to indicate this: k_n, ϵ_n, $\rho_{0,n}$ and T_n. Typically, we force ϵ_n to go to zero as n grows. We demonstrate a scheme for choosing the relay nodes and analyze the throughput performance of this scheme. Thus, we give an achievability result for T_n. We now state this result.

2.4 Main Result

Theorem 2.1. *Consider a network on n nodes whose edge strengths are drawn i.i.d. from a probability distribution function $f_n(\gamma)$. Let $F_n(\gamma)$ denote the cumulative distribution function corresponding to $f_n(\gamma)$ and define $Q_n(\gamma) = 1 - F_n(\gamma)$. Choose any β_n such that $Q_n(\beta_n) = \frac{\log n + \omega_n}{n}$, where $\omega_n \to \infty$ as $n \to \infty$. Then there exists a positive constant α such that a throughput of*

$$T = (1-\epsilon_n)\ \alpha k_n(\beta_n)\frac{\log(nQ_n(\beta_n))}{\log n}\ \log\left(1+\frac{a_n\beta_n}{\frac{\sigma^2}{P}+(k_n(\beta_n)-1)\mu_\gamma}\right) \tag{2.4}$$

is achievable for any positive a_n such that $a_n \leq 1$ and any $k_n(\beta_n)$ that satisfy the conditions:

1. $$k_n(\beta_n) \leq \alpha n\frac{\log(nQ_n(\beta_n))}{\log n} \tag{2.5}$$

2.

$$\epsilon_n \leq \frac{a_n^2}{\alpha(1-a_n)^2} \frac{(k_n(\beta_n)-1)\sigma_\gamma^2}{(\frac{\sigma^2}{P}+(k_n(\beta_n)-1)\mu_\gamma)^2} \frac{\log n}{\log(nQ_n(\beta_n))} \to 0 \qquad (2.6)$$

where μ_γ and σ_γ^2 are the mean and variance of γ respectively. The SINR threshold ρ_0 is given by $\frac{a_n\beta_n}{\frac{\sigma^2}{P}+(k_n(\beta_n)-1)\mu_\gamma}$.

The parameter β_n satisfying $Q_n(\beta_n) = \frac{\log n+\omega_n}{n}$ is the goodness threshold mentioned in Section 2.1.1. By figuratively drawing an edge when $\gamma > \beta_n$, we obtain a random graph that fits the well-studied model $\mathcal{G}(n,p)$. Condition (2.5) is needed to obtain a non-colliding schedule in this random graph. This issue is discussed in detail in Section 2.5. Once the schedule is obtained, we incorporate the effects of interference between non-colliding transmissions and provide an error analysis in Section 2.6. Condition (2.6) forces ϵ_n to go to zero. In Section 2.7 we combine the results of Sections 2.5 and 2.6 to prove the theorem. Note that the theorem indicates an achievable throughput and does not preclude that higher throughputs are possible.

Although it is not evident from the theorem statement, it turns out that the optimum number of hops h grows at most logarithmically with n. The throughput therefore depends most strongly on the number of simultaneous transmissions k_n and the SINR threshold ρ_0.

The throughput expression (2.4) is general and accommodates an arbitrary $f_n(\gamma)$. The parameter k_n is the number of non-colliding simultaneous transmissions. We discuss the constant α and the parameter a_n later. The joint selection of β_n, k_n, and a_n that maximizes the achievable throughput (2.4) is not easily expressed in closed-form as a function of the pdf $f_n(\gamma)$. In general, these parameters need to be determined on a case-by-case basis. We show how to find the necessary parameters in Section 2.8 where we give several examples.

Since (2.4) holds for any k_n satisfying (2.5), we may choose k_n as large as possible (achieving equality in (2.5)) and optimize only over a_n and β_n. In fact, when $\frac{\sigma^2}{P}-\mu_\gamma \geq 0$, it is possible to show that the optimum k_n is the maximum possible. We hence state a more specific achievability result.

Corollary 2.2. *In the network of Theorem 2.1, if* $\frac{\sigma^2}{P} - \mu_\gamma \geq 0$ *the throughput (2.4) is maximized by choosing* k_n *as large as possible.*

At this point we would like to refer back to the problem setting of [35] and note that their model of a random network, where nodes wish to send information at the rate of $\lambda(n)$ bits per second to a randomly chosen destination is closest to the problem we consider here. For the random network, an aggregate throughput capacity of $O(\sqrt{n/\log n})$ is obtained in [35]. (This is only slightly worse than the transport capacity of $O(\sqrt{n})$ for the somewhat different model of arbitrary networks, which has been discussed in the introduction to this work.) In the example presented in Section 2.8.2 we examine the scaling behavior of the throughput with a pdf $f_n(\gamma)$ that is obtained based on a distance-decay law. The effects of doing away with the geometric model become more clear with that example.

2.5 Scheduling Transmissions

With a view to meeting a minimum SINR of ρ_0 at every relay node at every hop, we impose the condition that each transmitting link be stronger than some threshold β_n. We require that $\gamma_{r_{i,t},r_{i,t+1}} \geq \beta_n$, where β_n is a design parameter. We denote links that satisfy $\gamma_{i,j} \geq \beta_n$ as *good*. We require the path from s_i to d_i to use only good links.

The threshold β_n is a parameter that we may choose as a compromise between quantity and quality of the connections. By making β_n large we increase the quality of the link. However, if we make it too large we risk not being able to form an uninterrupted path of good links from the source to the destination. In this section, we determine the relation between β_n and the lengths of source-destination paths.

Define $p_n = \mathrm{P}(\gamma \geq \beta_n)$ (for convenience, we drop the subscript n in the rest of this section). Using our wireless communication network, we define a graph on n vertices as follows: For (distinct) vertices i and j of the graph, draw an edge (i, j) if and only if $\gamma_{i,j} \geq \beta_n$ in the network. Call the resulting graph $G(n, p)$. The graph $G(n, p)$ then becomes an instance of a model called $\mathcal{G}(n, p)$ on n vertices in which edges are chosen independently and with probability p [26]. This graph shows the possible paths from

the various sources to the various destinations using only good links, but does not show the possible interference encountered if these paths are used simultaneously. We examine this interference in Section 2.6.

Graphs taken from the model $\mathcal{G}(n,p)$ have many known properties. For instance, the values of p for which the graph is connected is well-characterized. As p increases, the probability that the graph is connected goes to one. If $p = \frac{\log n + c + o(1)}{n}$ (where $c > 0$ need not be a constant) the probability of the graph being connected is $e^{-e^{-c}}$ [26]. This implies that there is a phase transition in the graph around $p = \frac{\log n}{n}$. For p less than this the probability of connectivity goes to zero rapidly and for p greater than this it goes to one rapidly. Another property that is well-studied is the *diameter*. The diameter of a graph is defined as the maximum distance between any two vertices of the graph, where the distance between two vertices is the minimum number of edges one has to traverse to go from one to the other. Results in [26] and [42] tell us that for p in the range of connectivity the diameter behaves like $\frac{\log n}{\log np}$. (It is also known that the average distance between two nodes has the same behavior.) This tells us that a message can be transmitted from one node to another using at most $\frac{\log n}{\log np}$ hops. What it leaves unanswered is the question of how to establish k such transmissions simultaneously and on non-colliding paths.

The problem of obtaining a non-colliding schedule can be thought of more generally as a problem of avoiding or reducing interference. Not surprisingly, several works that study throughput scaling in large networks encounter the same issue, sometimes for other network models. For instance, in [35] the number of routes that pass through a certain small area of the network (which they call a *cell*) can be thought of as the bottleneck that determines the overall throughput. Similarly, in [34], the number of disjoint paths that can be found in a certain area can be perceived as the limiting factor. Various techniques are used in these works to enable this calculation. While [35] uses results relating to the Vapnik-Chervonenkis dimension, [34] uses ideas inspired by percolation theory and random geometric graphs [53]. In the setting of this work, it is most natural to use random graph theory and we use a relatively recent result regarding vertex-disjoint paths by Broder et al [43] in order to find a satisfactory

non-colliding schedule.

2.5.1 Scheduling using Vertex-Disjoint Paths in $G(n,p)$

Two paths that do not share a vertex are called vertex-disjoint. Note that any two paths that are vertex-disjoint satisfy our "no-collisions" property; however, the reverse statement is not true. Thus, the vertex-disjoint condition is stronger than our requirement of non-colliding paths. For a set of k (disjoint) pairs of vertices (s_i, d_i), the question of whether there exists a set of vertex-disjoint paths connecting them is addressed in [43]. Their result states that with high probability, for every (sufficiently random) set of k pairs (s_i, d_i) and k not greater than $\alpha_1 n \frac{\log np}{\log n}$, where α_1 is a constant, there exists a set of vertex-disjoint paths. This result is within a constant factor of the best one can hope to achieve since the average distance between nodes in $\mathcal{G}(n,p)$ is $\frac{\log n}{\log np}$, and thus we can certainly have no more than $n\frac{\log np}{\log n}$ vertex-disjoint paths. Also stated in [43] is an algorithm that finds k paths using various random walk and flow techniques. Here we reproduce their main result.

Theorem 2.3. *Suppose that $G = G(n,p)$ and $p \geq \frac{\log n + \omega_n}{n}$, where $\omega_n \to \infty$. Then there exist two positive constants α_1, α_2 such that, with probability approaching one, there are vertex-disjoint paths connecting s_i to d_i for any set of pairs*

$$F = \{(s_i, d_i) | s_i, d_i \in \{1, \ldots, n\}, i = 1, \ldots, k\}$$

satisfying

1. *The pairs in F for $i = 1, \ldots, k$ are disjoint.*
2. *The total number of pairs, $k = |F|$, is not greater than $\alpha_1 n \frac{\log np}{\log n}$;*
3. *For every vertex $v \in \{1, \ldots, n\}$, no more than an α_2-fraction of its set of neighbors, $N(v)$, are prescribed endpoints, that is $|N(v) \cap (S \cup D)| \leq \alpha_2 |N(v)|$, where $S = \{s_i\}$ and $D = \{d_i\}$.*

Furthermore, these paths can be constructed by an explicit randomized algorithm in polynomial time.

In fact, the existence of the paths is proved by stating and analyzing a randomized algorithm that finds them. However, we use this theorem only as an existence result to demonstrate achievable throughputs. Some comments about their randomized algorithm can be found in Sections 2.6 and 2.10.1.

In our communication network, Condition 1 that (s_i, d_i) be disjoint pairs is already met. The second condition imposes a restriction on how large k can be. Since the k source-destination pairs are chosen at random, the third condition is also met. (In fact, the third condition is imposed in [43] to prevent someone from choosing the (s_i, d_i) pairs in a particularly adversarial manner using knowledge of the graph structure.)

We can restate the theorem for our purposes.

Theorem 2.4. *Suppose that $G = G(n, p)$ and $p \geq \frac{\log n + \omega_n}{n}$, where $\omega_n \to \infty$. Then there exists a constant $\alpha > 0$ such that, with probability approaching one, there are vertex-disjoint paths connecting s_i to d_i for any set of disjoint, randomly chosen source-destination pairs*

$$F = \{(s_i, d_i) | s_i, d_i \in \{1, \ldots, n\}, i = 1, \ldots, k\}$$

provided $k = |F|$ is not greater than $\alpha n \frac{\log np}{\log n}$.

The constant α in this theorem is the same α required in Theorem 2.1. It is not explicitly specified. We examine the lengths that these k paths can have in the following lemma.

Lemma 2.5. *Almost all of the $k = \alpha n \frac{\log np}{\log n}$ vertex-disjoint paths obtainable under Theorem 2.4 have lengths that grow no faster than $\frac{\log n}{\alpha \log np}$.*

Proof. Suppose that some fraction of paths, say $c_n k$ where $c_n > 0$ have average lengths of the form $\frac{\log n}{\log np}(1 + \omega'_n)$ where ω'_n goes to infinity. Since there are n nodes in the

network, we have

$$n \geq c_n k \times \frac{\log n}{\log np}(1+\omega_n') = c_n \alpha n \frac{\log np}{\log n} \times \frac{\log n}{\log np}(1+\omega_n') = c_n \alpha n (1+\omega_n').$$

This implies that $1 \geq \alpha c_n(1+\omega_n')$ and therefore c_n must go to zero. Therefore we conclude that at most a vanishing fraction of the k paths can have lengths that grow faster than $\frac{\log n}{\log np}$ and, asymptotically, all the paths have lengths that grow no faster than $\frac{\log n}{\alpha \log np}$. □

Hence the number of hops h is (asymptotically) at most $\frac{\log n}{\alpha \log np}$. We use this fact in the error analysis in the following section.

2.6 Probability of Error

Consider a schedule of $k \leq \alpha n \frac{\log np}{\log n}$ non-colliding paths. Theorem 2.4 shows that such a schedule exists. One possible (but often impractical) way to obtain such a schedule is to use an exhaustive search that first lists all the paths between every source-destination pair and then randomly chooses a set that satisfies the vertex-disjoint property. Because we thereby choose a path based on vertices rather than edges, we are assured that any edges that might exist between vertices along one path to vertices along another are i.i.d., Bernoulli distributed with parameter p. We also conclude that the channel connections between nodes along different paths in the network are i.i.d. with distribution $f_n(\gamma)$.

More generally, randomized algorithms that choose non-colliding paths without using edge information between such paths also have the property of generating i.i.d. interference between the paths. An example of such a randomized algorithm that avoids an exhaustive search is [43].

We now consider the probability that a particular message fails to reach its intended destination. Destination d_i fails to receive message M_i if the SINR falls below ρ_0 at any of the h relay nodes $r_{i,1}, \ldots, r_{i,h} = d_i$. Denote by E_t the event that relay node $r_{i,t}$ has SINR greater than ρ_0. Note that the events $E_1, \ldots, E_h$ are identically

distributed. Therefore we have,

$$\mathrm{P}(M_i \text{ is received successfully}) = \mathrm{P}(\bigcap_{t=1}^{h} E_t) = 1-\mathrm{P}(\bigcup_{t=1}^{h} \sim E_t) \geq 1-\sum_{t=1}^{h} \mathrm{P}(\sim E_t) = 1-h\mathrm{P}(\sim E_1) \tag{2.7}$$

where the inequality comes from the union bound. We now compute $\mathrm{P}(\sim E_1)$. This is the event that node $r_{i,1}$ has an SINR lower than ρ_0.

$$\begin{aligned}
\mathrm{P}(\sim E_1) &= \mathrm{P}(\rho_{r_{i,1}} \leq \rho_0) \\
&= \mathrm{P}\left(\frac{P\gamma_{s_i,r_{i,1}}}{\sigma^2 + P\sum_{j\neq i}\gamma_{s_j,r_{i,1}}} \leq \rho_0\right) \\
&= \mathrm{P}\left(\sum_{j\neq i}\gamma_{s_j,r_{i,1}} \geq \frac{P\gamma_{s_i,r_{i,1}} - \rho_0\sigma^2}{P\rho_0}\right) \\
&\leq \mathrm{P}\left(\sum_{j\neq i}\gamma_{s_j,r_{i,1}} \geq \frac{P\beta_n - \rho_0\sigma^2}{P\rho_0}\right) \\
&= \mathrm{P}\left(\frac{1}{k-1}\sum_{j\neq i}\gamma_{s_j,r_{i,1}} - \mu_\gamma \geq \frac{P\beta_n - \rho_0\sigma^2}{(k-1)P\rho_0} - \mu_\gamma\right) \\
&\leq \mathrm{P}\left(\left|\frac{1}{k-1}\sum_{j\neq i}\gamma_{s_j,r_{i,1}} - \mu_\gamma\right| \geq \frac{P\beta_n - \rho_0\sigma^2}{(k-1)P\rho_0} - \mu_\gamma\right) \\
&\leq \frac{\sigma_\gamma^2/(k-1)}{(\frac{P\beta_n-\rho_0\sigma^2}{(k-1)P\rho_0} - \mu_\gamma)^2}
\end{aligned} \tag{2.8}$$

where the first inequality follows because $\gamma_{s_i,r_{i,1}} \geq \beta_n$ and (2.8) comes from the Chebyshev inequality and the fact that the variance of $\frac{1}{k-1}\sum_{j\neq i}\gamma_{s_j,r_{i,1}}$ is $\sigma_\gamma^2/(k-1)$. The second inequality requires the condition $\frac{P\beta_n-\rho_0\sigma^2}{(k-1)P\rho_0} - \mu_\gamma \geq 0$, or

$$\rho_0 \leq \frac{\beta_n}{\frac{\sigma^2}{P} + (k-1)\mu_\gamma}. \tag{2.9}$$

This condition on ρ_0 is intuitively satisfying: if we assume that k is large, then we expect the interference term in the denominator of the SINR to be approximately $(k-1)\mu_\gamma$. This would imply that setting the threshold ρ_0 to less than $\frac{\beta_n}{\frac{\sigma^2}{P}+(k-1)\mu_\gamma}$ would be sufficient to ensure that most hops would exceed this threshold.

Note that in the above analysis for $\mathrm{P}(\rho_{r_i,t} \leq \rho_0)$, we have assumed that there are $(k-1)$ interference terms. This would be true if all k messages are transmitted in that particular time slot. However, this may not be the case, if some of the messages have already reached their destinations successfully by that time or have already failed to be decoded at some relay node. In such a case, there will be fewer than $(k-1)$ interference terms. This means that the calculation above is conservative and the actual probability of error may be smaller than that obtained above. However, from the relevant theory involving random graphs as well as from the simulations, we expect the path lengths to fall in a narrow range of values. Thus, most messages reach their destination within very few time slots of each other. Therefore, we believe that the above error analysis is not too conservative and hence do not expect a significantly lower error probability in practice.

We define ϵ_n to be the probability of failing to meet the SINR threshold along one or more of the hops. From (2.7), $\epsilon_n \leq h\mathrm{P}(\sim E_1)$. We force $h\mathrm{P}(\sim E_1)$ to go to zero. From Lemma 2.5, h is at most $\frac{\log n}{\alpha \log np}$ and we have

$$\epsilon_n \leq h\mathrm{P}(\sim E_1) \leq \frac{\log n}{\alpha \log np} \frac{\sigma_\gamma^2}{(k-1)(\frac{P\beta_n - \rho_0\sigma^2}{(k-1)P\rho_0} - \mu_\gamma)^2} \tag{2.10}$$

and we require the right-hand side to go to zero.

The inequality (2.10) requires γ to have a variance that does not go to infinity. There are several distributions of practical interest in which the variance does go to infinity, but the mean is finite. (For example, $f(\gamma) = \frac{c}{(1+\gamma)^m}$ for $m > 2$ is considered in [52].) In this case, an alternative inequality can be obtained by applying the Markov bound to $\mathrm{P}(\sim E_1)$ rather than the Chebyshev bound. The result is

$$\epsilon_n \leq h\mathrm{P}(\sim E_1) \leq \frac{\log n}{\alpha \log np} \frac{(k-1)\mu_\gamma P\rho_0}{P\beta_n - \rho_0\sigma^2}. \tag{2.11}$$

An achievable throughput can be obtained using either the Chebyshev bound of (2.10) or the Markov inequality above. Theorem 2.1 is obtained using the Chebyshev inequality. Theorem 2.6, presented at the end of Section 2.7, is an achievability result

obtained using the Markov inequality (2.11). In general, we expect the Chebyshev inequality to be tighter than the Markov inequality and therefore prefer to use Theorem 2.1 whenever γ has finite variance.

2.7 Proof of Theorem 2.1

We now combine the results of Section 2.5 on the maximum number of non-colliding paths and Section 2.6 on the probability of successful transmission along these paths. We need $p = \mathrm{P}(\gamma \geq \beta_n) = Q_n(\beta_n) = \frac{\log n + \omega_n}{n}$ in order to do scheduling. In addition, we need:

1. To have non-colliding paths (Theorem 2.4),

$$k \leq \alpha n \frac{\log np}{\log n}.$$

2. To meet the SINR threshold (equation (2.10)),

$$\epsilon_n \leq \frac{\log n}{\alpha \log np} \frac{\sigma_\gamma^2}{(k-1)(\frac{P\beta_n - \rho_0\sigma^2}{(k-1)P\rho_0} - \mu_\gamma)^2} \to 0.$$

3. To apply the Chebyshev inequality (equation (2.9)),

$$\rho_0 \leq \frac{\beta_n}{\frac{\sigma^2}{P} + (k-1)\mu_\gamma}.$$

To satisfy the third condition above we set

$$\rho_0 = \frac{a_n \beta_n}{\frac{\sigma^2}{P} + (k-1)\mu_\gamma}$$

where $0 \leq a_n \leq 1$. Substituting for this in the second condition, we get

$$\epsilon_n \leq \frac{a_n^2}{\alpha(1-a_n)^2} \frac{(k_n(\beta_n)-1)\sigma_\gamma^2}{(\frac{\sigma^2}{P} + (k_n(\beta_n)-1)\mu_\gamma)^2} \frac{\log n}{\log(nQ_n(\beta_n))} \to 0.$$

This and the first condition above are the only conditions on k. For any k satisfying these two conditions we get an achievable throughput. This gives us Theorem 2.1.

The theorem gives an achievable throughput as a function of β_n, a_n and k_n but does not attempt to optimize these parameters. Because ϵ_n goes to zero and h is determined by β_n, to find the optimum k we need to maximize $k\log(1+\rho_0) = k\log(1+\frac{a_n\beta_n}{\frac{\sigma^2}{P}+(k-1)\mu_\gamma})$ over k. In the particular case when $\frac{\sigma^2}{P}-\mu_\gamma$ is positive, the expression is non-decreasing in k (the first derivative is non-negative). Hence satisfying (2.5) with equality is optimum. This proves Corollary 2.2.

Finally, we state without proof an achievability result obtained using the Markov inequality (2.11) to bound the error, rather than the Chebyshev inequality (2.10). This result can be used in place of Theorem 2.1 for distributions that have a finite mean but an infinite variance.

Theorem 2.6. *Consider a network on n nodes whose edge strengths are drawn i.i.d. from a probability distribution function $f_n(\gamma)$. Let $F_n(\gamma)$ denote the cumulative distribution function corresponding to $f_n(\gamma)$ and define $Q_n(\gamma) = 1 - F_n(\gamma)$. Choose any β_n such that $Q_n(\beta_n) = \frac{\log n+\omega_n}{n}$, where $\omega_n \to \infty$ as $n \to \infty$. Then there exists a positive constant α such that a throughput of*

$$T = (1-\epsilon_n)\ \alpha k_n(\beta_n)\frac{\log(nQ_n(\beta_n))}{\log n}\ \log\left(1+\frac{\beta_n}{\frac{\sigma^2}{P}+b_n(k_n(\beta_n)-1)\mu_\gamma}\right) \tag{2.12}$$

is achievable for any positive b_n such that $b_n \geq 1$ and any $k_n(\beta_n)$ that satisfy the conditions:

1. $$k_n(\beta_n) \leq \alpha n\frac{\log(nQ_n(\beta_n))}{\log n}. \tag{2.13}$$

2. $$\epsilon_n \leq \frac{1}{\alpha b_n}\frac{\log n}{\log(nQ_n(\beta_n))} \to 0. \tag{2.14}$$

where μ_γ is the mean of γ. The SINR threshold is $\rho_0 = \frac{\beta_n}{\frac{\sigma^2}{P}+b_n(k_n(\beta_n)-1)\mu_\gamma}$.

2.8 Examples and Applications

In this section we apply Theorem 2.1 to some particular channel distributions. Since, as in geometric models, the throughput is often interference-limited, we find that densities that lead to significant interference per transmitter generally underperform those that generate only a small amount of interference.

2.8.1 Shadow-Fading Model

We revisit the model (2.1)

$$f_n(\gamma) = (1 - p_n)\delta(\gamma) + p_n\delta(\gamma - 1) \tag{2.15}$$

where $\delta(\cdot)$ is the Dirac delta-function. This pdf models the situation where strong shadow-fading is present. The signal power is zero in the presence of an obstruction and is one otherwise. We find the value of p that maximizes the throughput. (We drop the subscript n.) A natural choice for the goodness threshold β_n is 1, which gives $Q(\beta) = p$. We need to satisfy $p \geq (\log n + \omega_n)/n$ (where $\omega_n \to \infty$) in order to use Theorem 2.1.

Note that we have $\mu_\gamma = p$ and $\sigma_\gamma^2 = p(1 - p)$. It is possible to check that unless $p \to 0$, the throughput is at most constant. With $p \to 0$ and sufficiently large n, the condition $\frac{\sigma^2}{P} - \mu_\gamma = \frac{\sigma^2}{P} - p \geq 0$ is satisfied. Therefore, according to Corollary 2.2 the maximum possible k achieves maximum throughput. Hence we consider $k = \alpha n \frac{\log np}{\log n}$. Since $p = \frac{\log n + \omega_n}{n}$, $k \to \infty$ and we may replace $k - 1$ by k in (2.6) and the SINR threshold. Since kp also goes to infinity, (2.6) becomes $\epsilon_n \leq \frac{a_n^2}{\alpha^2(1-a_n)^2} \frac{\log^2 n}{\log^2(np)} \frac{1}{n} \to 0$. Therefore a_n may be any positive constant $a < 1$. With this, the SINR threshold becomes $\rho_0 = \frac{a}{\frac{\sigma^2}{P} + \alpha np \frac{\log np}{\log n}} \approx \frac{a}{\alpha np \frac{\log np}{\log n}}$ which goes to zero. Thus $\log(1 + \rho_0) \approx \rho_0$ and we have $\frac{k}{h} \log(1 + \rho_0) = \frac{a\alpha}{p} \frac{\log np}{\log n}$. This is maximized when p is as small as possible, or $p = \frac{\log n + \omega_n}{n}$. The result is summarized in the corollary below.

Corollary 2.7. *Consider a network on n nodes where edge strengths are drawn i.i.d.*

from the distribution in (2.15). Then for large n the maximum throughput is

$$T = \left(1 - \frac{a^2}{\alpha^2(1-a)^2}\frac{\log^2 n}{\log^2(\log n + \omega_n)}\frac{1}{n}\right) a\alpha\frac{\log(\log n + \omega_n)}{(\log n + \omega_n)\log n}\ n$$

as $n \to \infty$, which is achieved when $p = \frac{\log n + \omega_n}{n}$ and where ω_n is any function going to infinity and $0 < a < 1$ and $\alpha < 1$ are constants.

This throughput is almost linear in n and requires the network to be sparsely connected; with a connection probability of $(\log n)/n$, each node is connected with only approximately $\log n$ other nodes. For example with $n = 1000$ nodes, we have $(\log n)/n = 0.0069$ and each node connects on average to only seven other nodes. Perhaps surprisingly, increasing or decreasing this connectivity has a detrimental effect. While it is clear that it is possible for a network to be under-connected, it is apparently also possible for a network to be over-connected. The simulations in Section 2.10.2 also demonstrate this effect.

2.8.1.1 Implications for a Certain Radio Model

In [31, 32] a wireless connectivity model is introduced where the probability of a good link is expressed as

$$p(\hat{r}) = \frac{1}{2}\left[1 - \text{erf}\left(3.07\frac{\log \hat{r}}{\xi}\right)\right] \tag{2.16}$$

where $\hat{r}$ is a (suitably normalized) distance between the transmitter and receiver and ξ is a parameter that depends on the degree of shadow fading and the distance pathloss exponent. Usually $\xi \in [0, 6]$ where large values indicate a strong shadow component. The links between different sources or destinations are modeled as statistically independent.

For nodes approximately $\hat{r}$ from each other, the model (2.16) is equivalent to our model of shadow-fading (2.15) with $p = p(\hat{r})$. As we show in Section 2.8.1, maximum throughput is attained for $p \approx (\log n)/n$. The "equivalent distance" for nodes is found by solving

$$p = \frac{\log n}{n} = \frac{1}{2}\left[1 - \text{erf}\left(3.07\frac{\log \hat{r}}{\xi}\right)\right] \tag{2.17}$$

for $\hat{r}$. Nodes approximately this distance from each other then have the excellent throughput promised in Corollary 2.7. Because we cannot have a large network of nodes exactly equidistant from each other, equation (2.17) has operational meaning only if the link probability is relatively insensitive to the distance $\hat{r}$ when $p \approx (\log n)/n$. We show that it is.

As the number of nodes n increases, the optimal link-probability $(\log n)/n$ decreases, or, equivalently, the distance $\hat{r}$ between nodes increases. For large $\hat{r}$, we may approximate $\frac{1}{2}(1 - \operatorname{erf} x) \approx 1/(2\sqrt{\pi}x)\exp(-x^2)$, and thus (2.17) becomes

$$p = \frac{\log n}{n} = \frac{\xi}{10.88 \log \hat{r}} e^{-3.07 \log^2 \hat{r}/\xi}.$$

The sensitivity of p as a function of $\hat{r}$ is very low when p is small. We show this in Figure 2.3, where we display p versus $\hat{r}$ for various values of ξ. The dotted lines in the figure shows the approximate optimal operating point p for networks with 100 and 1000 nodes. We see that the optimal p is generally very small and relatively insensitive to $\hat{r}$, and therefore the best network performance is generally obtained when the nodes are relatively far apart, with a wide range of acceptable distances. This suggests that a large high-throughput network of nodes with optimal (small) p is possible.

We comment that the authors in [32] also consider how shadow fading can reduce the hop-count in a network and they use some graph-theoretic concepts in their arguments. They do not, however, attempt to obtain a throughput result by finding simultaneous non-colliding paths, nor do they incorporate the detrimental effects of interference to show that a network can be "too connected."

2.8.2 Density obtained from a Decay Law

In this example we construct a pdf from the marginal density of the channel strengths in a geometric model. For every node, the channel coefficients to the remaining nodes follow a deterministic law based on distance. If we group these coefficients according to their magnitude γ, we obtain a certain number of coefficients whose magnitude

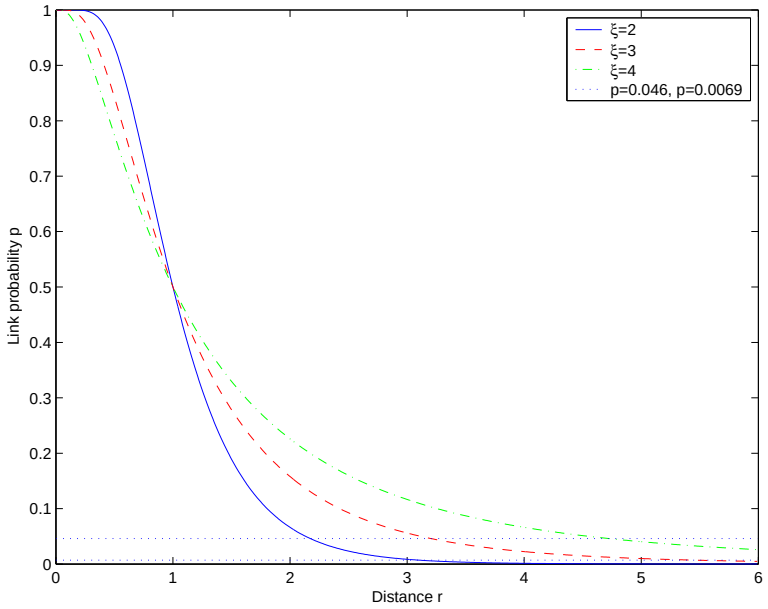


Figure 2.3: Link probability p versus distance $\hat{r}$ as given by (2.17) for $\xi = 2, 3, 4$. Also shown are dotted lines at $p = (\log 100)/100 \approx 0.046$ and $p = (\log 1000)/1000 \approx 0.0069$ indicating the optimum throughput point for shadow-fading with 100 and 1000 nodes, respectively. As a function of $\hat{r}$, p is relatively insensitive for large $\hat{r}$.

falls in the interval $(\gamma, \gamma + d\gamma)$. We seek a pdf whose average number of magnitudes matches this deterministic law.

In an actual geometric model the distribution of channel magnitudes depends on the location of the nodes. We make a simplifying assumption: We suppose that the nodes are in a circular disk and consider the node at the center of the disk to derive the density. We thereby ignore the effects of the disk boundary. We assume the nodes are dropped with density Δ (nodes per unit area) but ensuring a minimum distance of d from the center. The area of the entire disk is n/Δ.

In deriving the density of the channel coefficients, we use a power law $g(r)$, where a node transmitting with power P is received by another node at distance r with power $Pg(r)$. We assume that $g(\cdot)$ is monotonically decreasing. The most significant difference between our model and the standard geometric model is that our channel coefficients are independent which does not happen in the geometric model. The geometric model has a correlation structure in the coefficients where channels of similar strength are clustered in rings around the center node. In our model, coefficients of

similar strength, although the same in number as the geometric model, are distributed randomly and are not necessarily geometrically colocated.

Consider a node at the center of the disk transmitting at power P. The fraction of nodes receiving power $\leq \gamma P$ is given by $1 - \frac{\Delta}{n} 2\pi((g^{-1}(\gamma))^2 - d^2)$ where $\gamma \in [g(\sqrt{\frac{n}{2\pi\Delta} + d^2}), g(d)]$ In particular, if we have a decay law of the form $g(r) = \frac{1}{r^m}$, the fraction of nodes receiving power $\leq \gamma P$ is given by

$$1 - \frac{\Delta}{n} 2\pi (\frac{1}{\gamma^{2/m}} - d^2)$$

for $\gamma \in [\left(\frac{2\pi\Delta}{n+2\pi\Delta d^2}\right)^{m/2}, \frac{1}{d^m}]$.

This is a cumulative distribution function and by differentiating it with respect to γ we obtain the pdf for the edge strengths seen by the central node as

$$f_n(\gamma) = \frac{4\pi\Delta}{nm} \frac{1}{\gamma^{1+\frac{2}{m}}}, \qquad \gamma \in \left[\left(\frac{2\pi\Delta}{n + 2\pi\Delta d^2} \right)^{m/2}, \frac{1}{d^m} \right], \qquad m > 0. \tag{2.18}$$

We assume that connections are drawn i.i.d. from this distribution.

We apply our results to this network and obtain the following corollary.

Corollary 2.8. *Consider a network on n nodes where edge strengths are drawn i.i.d. from the distribution*

$$f_n(\gamma) = \frac{4\pi\Delta}{nm} \frac{1}{\gamma^{1+\frac{2}{m}}}, \qquad \gamma \in \left[\left(\frac{2\pi\Delta}{n + 2\pi\Delta d^2} \right)^{m/2}, \frac{1}{d^m} \right], \qquad m > 0$$

Then the following values of ϵ_n and throughputs are achievable:

$$\epsilon_n \leq \begin{cases} \frac{a^2}{\alpha^2(1-a)^2} (\frac{(2-m)^2}{4(1-m)} - 1) \frac{\log^2 n}{\log^2(\log n + \omega_n)} \frac{1}{n} & m < 1 \\ \frac{a^2}{4(1-a)^2} \frac{\log^3 n}{\log^2(\log n + \omega_n)} \frac{1}{\alpha^2 n} & m = 1 \\ \frac{a^2 (2\pi\Delta)^{1-m} (2-m)^2}{4(1-a)^2 (m-1) d^{2(m-1)}} \frac{\log^2 n}{\log^2(\log n + \omega_n)} \frac{1}{\alpha^2 n^{2-m}} & 1 < m < 2 \\ \frac{a^2}{2\pi\Delta(1-a)^2 d^2} \frac{1}{\alpha^2 \log^2(\log n + \omega_n)} & m = 2 \\ \frac{1}{w_n^2} \frac{2\pi\Delta P^2}{(m-1) d^{2(m-1)} \alpha \sigma^4} & m > 2 \end{cases} \tag{2.19}$$

$$T = \begin{cases} (1-\epsilon_n)\frac{a(2-m)\alpha}{2}\frac{\log(\log n+\omega_n)}{\log n(\log n+\omega_n)^{m/2}}\ n^{m/2} & m<1 \\ (1-\epsilon_n)\frac{a\alpha}{2}\frac{\log(\log n+\omega_n)}{\log n(\log n+\omega_n)^{1/2}}\ n^{1/2} & m=1 \\ (1-\epsilon_n)\frac{a(2-m)\alpha}{2}\frac{\log(\log n+\omega_n)}{\log n(\log n+\omega_n)^{m/2}}\ n^{m/2} & 1<m<2 \\ (1-\epsilon_n)a\alpha\frac{\log(\log n+\omega_n)}{\log^2 n(\log n+\omega_n)}\ n & m=2 \\ (1-\epsilon_n)\frac{P\alpha^2(2\pi\Delta)^{m/2}}{\sigma^2 w_n}\frac{\log^2(\log n+\omega_n)}{\log^2 n(\log n+\omega_n)^{m/2}}\ n & m>2, \end{cases} \tag{2.20}$$

where $a<1$ and $\alpha<1$ are constants and ω_n and w_n are functions going to infinity.

This corollary gives almost linear throughput for $m \geq 2$. This differs substantially from the $O(\sqrt{n})$ or $O(\sqrt{n/\log n})$ results obtained for the structured deterministic model with the same decay law. Our results show that it is not the marginal distribution of the power that impedes the throughput in a geometric power-decay network, but rather the spatial distribution of these powers. We notice that in the geometric model, nodes transmit to their nearest neighbors and therefore messages take up to $\sqrt{n}$ hops to reach their intended destinations. In the random model, nodes talk across their good links and only $\log n$ hops are necessary to send a message across. This is due to two factors: the first is that far fewer nodes get drowned out in the interference when one node transmits, thus permitting more nodes to transmit simultaneously, and the second is that any two nodes in the random graph are only $\log n$ hops away, rather than $\sqrt{n}$ hops as in the deterministic model.

It is easy to see that both advantages come about from the absence of the geometric constraints. If we think of the example of this section in terms of the network of [35], but with the spatial constraints removed, it is not surprising that the throughput scaling is much more favorable than $O(\sqrt{n/\log n})$. This improvement arises from the introduction of randomness in the network model.

2.8.3 A Distribution with Constant Mean and Variance

Consider a general distribution $f_n(\gamma)$ that has a constant mean and variance. For such a distribution, one can show that choosing $k \to \infty$ is the best choice and leads to the following corollary.

Corollary 2.9. *Consider a network on n nodes where edge strengths are drawn i.i.d. from a distribution $f_n(\gamma)$ where the mean μ_γ and variance σ_γ^2 of γ are independent of n. Then the throughput is given by*

$$T = \left(1 - \frac{a^2\sigma_\gamma^2}{\alpha^2(1-a)^2\mu_\gamma^2}\frac{\log^2 n}{\log^2(nQ_n(\beta_n))}\frac{1}{n}\right)\frac{a\alpha}{\mu_\gamma}\frac{\beta_n \log(nQ_n(\beta_n))}{\log n}$$

and the optimum β_n maximizes $\beta_n \log(nQ_n(\beta_n))$ while satisfying $Q_n(\beta_n) \geq \frac{\log n + \omega_n}{n}$.

Perhaps surprisingly, distributions with constant mean and variance, while allowing us to apply Corollary 2.9, can have widely different throughputs. This is illustrated by the next few examples.

2.8.4 An Exponential Density

Let $f_n(\gamma) = e^{-\gamma}$. For this pdf, the mean and variance are constant, and we can apply Corollary 2.9. The obtained throughput is summarized below.

Corollary 2.10. *Consider a network on n nodes where edge strengths are drawn i.i.d. from a distribution $f_n(\gamma) = e^{-\gamma}$. Then a throughput of*

$$T = \left(1 - \frac{a^2}{\alpha^2(1-a)^2}\frac{4}{n}\right)\frac{a\alpha \log n}{4}$$

is achievable as $n \to \infty$ where $\alpha < 1$, $a < 1$ are constants.

The throughput grows only logarithmically with n. This network has good connectivity since the number of hops is small, but is also unfortunately dominated by interference. Thus, few transmissions can occur simultaneously. We show in Section 2.9 that this throughput is tight to first-order in n.

2.8.5 A Heavy-Tail Distribution

Consider a network on n nodes where edge strengths are drawn i.i.d. from $f_n(\gamma) = \frac{c}{1+\gamma^4}, \gamma \geq 0$ where c is such that $f_n(\gamma)$ integrates to 1. The mean and variance of

$f_n(\gamma)$ are constant with respect to n. Therefore we apply Corollary 2.9. The optimal β_n equals $\frac{n^{1/3}}{(\log n+\omega_n)^{1/3}}\frac{c^{1/3}}{3^{1/3}}$.

Corollary 2.11. *Consider a network on n nodes where edge strengths are drawn i.i.d. from the distribution* $f_n(\gamma) = \frac{c}{1+\gamma^4}, \gamma \geq 0$. *The throughput is*

$$\begin{aligned} T &= \left(1 - \frac{a^2\sigma_\gamma^2}{\alpha^2\mu_\gamma^2(1-a)^2}\frac{\log^2 n}{\log^2(\log n+\omega_n)}\frac{1}{n}\right)\frac{a(c/3)^{1/3}\alpha}{\mu_\gamma}\frac{\log(\log n+\omega_n)}{\log n(\log n+\omega_n)^{1/3}}\ n^{1/3} \\ &\approx \frac{a(c/3)^{1/3}\alpha}{\mu_\gamma}\frac{\log\log n}{\log^{4/3} n}\ n^{1/3}. \end{aligned}$$

2.8.6 Lognormal Fading

Consider a network on n nodes where edge strengths are drawn from a lognormal distribution. Thus $f_n(\gamma) = \frac{1}{S\sqrt{2\pi}\gamma}\exp(-(\log\gamma - M)^2/2S^2), \gamma \geq 0$ where M and S are parameters of the distribution. We have $\mu_\gamma = e^{M+S^2/2}$ and $\sigma_\gamma^2 = e^{S^2+2M}(e^{S^2}-1)$. Because the mean and variance are constant, we may apply Corollary 2.9 and get the following result.

Corollary 2.12. *Consider a network on n nodes where edge strengths are drawn i.i.d. from the distribution* $f_n(\gamma) = \frac{1}{S\sqrt{2\pi}\gamma}\exp(-(\log\gamma - M)^2/2S^2), \gamma \geq 0$. *The throughput is then*

$$\begin{aligned} T &= \left(1 - \frac{a^2\sigma_\gamma^2}{\alpha^2\mu_\gamma^2(1-a)^2}\frac{\log^2 n}{\log^2(\log n+\omega_n)}\frac{1}{n}\right)\frac{a\alpha}{\mu_\gamma}\frac{e^M e^{\sqrt{2}S\sqrt{\log n}}\log(\log n+\omega_n)}{\log n} \\ &\approx \frac{a\alpha e^{M+\sqrt{2}S}}{\mu_\gamma}\frac{\log\log n}{\log n}\ e^{\sqrt{\log n}}. \end{aligned}$$

We see that the throughput grows as $e^{\sqrt{\log n}}$, which can also be written as $n^{\frac{1}{\sqrt{\log n}}}$ or $(\log n)^{\frac{\sqrt{\log n}}{\log\log n}}$. Thus the througput is considerably better than $\log n$ obtained with the exponential density (Rayleigh fading).

2.8.7 Tradeoff between k and ρ_0

In most of the examples above we notice that the optimal k goes to infinity; hence the optimal $\rho_0 = \frac{a\beta_n}{\frac{\sigma^2}{P}+(k-1)\mu_\gamma}$ goes to zero. In these cases we approximate $\log(1+\rho_0)$ by ρ_0. In addition, if $k\mu_\gamma$ goes to infinity, we can further approximate ρ_0 as $\frac{a\beta_n}{k\mu_\gamma}$. In this case, we have $\frac{k}{h}\log(1+\rho_0) \approx \frac{a\beta_n}{h\mu_\gamma}$. This expression depends only on β_n and is independent of k and ρ_0. We can therefore increase (decrease) k, thus decreasing (increasing) $\rho_0 = \frac{a\beta_n}{\frac{\sigma^2}{P}+(k-1)\mu_\gamma}$ and (as long as $k\mu_\gamma \to \infty$) the throughput remains unaffected. Hence it is sometimes possible to trade off the number of simultaneously communicating source-destination pairs with the SINRs at which they communicate without affecting the aggregate throughput.

2.9 Upper Bounds

Our method of finding the throughput relies on finding good edges along which the desired communication can take place. Other methods may do better. In the cases where the throughput is of the form $\frac{n}{\log^d n}$ the optimal throughput cannot be better by more than the factor $\log^d n$ because the maximal throughput cannot scale more than linearly (unless the channel density is somehow chosen such that the maximal received power increases as the number of nodes increases – we exclude such densities here).

However, when the throughput we compute turns out to be of the order of n^d for $d < 1$, or $\log n$ as with the exponential density, it is not clear that we cannot do better. In this section we present an approach to computing an upper bound on throughput that shows that we sometimes cannot do better.

The throughput is given by $(1-\epsilon)\frac{k}{h}\log(1+\rho_0)$. We ignore the h in the denominator and find an upper bound for $k\log(1+\rho_0)$. Thus, we allow ourselves to choose k source-destination pairs from a given network and find the highest SINR threshold that can be met for all of them simultaneously. This is equivalent to finding a bound for the best single hop communication. By doing this, our achievability results are certain to

be at least a factor of h away from the upper bound. However, we know that h can be no larger than $\frac{\log n}{\log(\log n+\omega_n)}$, which is often a small factor.

There are $\binom{n}{k}\binom{n-k}{k}k!$ ways of choosing k source-destination pairs in a network. Assume that a threshold ρ_0 is fixed. Then, for a randomly drawn set of source-destination pairs, there is a probability, say p_s, that a received message satisfies the SINR threshold and is decoded successfully. The probability that all k received messages satisfy the threshold is p_s^k. Therefore, for a given pair (k, ρ_0), the expected number of sets of k source-destination pairs that satisfy the threshold ρ_0 is

$$M_n(k, \rho_0) = \binom{n}{k}\binom{n-k}{k}k!p_s^k.$$

Note that p_s depends on ρ_0, k and the pdf $f_n(\gamma)$ from which the connections are drawn. We say that a (k, ρ_0) pair is feasible if there exists at least one set of k source-destination pairs such that each of the k SINRs exceeds ρ_0. We bound the probability that a particular (k, ρ_0) pair is feasible as.

$$\begin{aligned}
\mathrm{P}((k,\rho_0) \text{ is feasible}) &= \mathrm{P}(\#\text{ of } k\text{-pairs that satisfy the threshold } \rho_0 \text{ is} \geq 1) \\
&\leq \mathrm{E}(\#\ k\text{-pairs that satisfy the threshold } \rho_0) \\
&= M_n(k,\rho_0)
\end{aligned}$$

where the Markov inequality is used.

If for a particular choice of (k, ρ_0) we have $M_n(k, \rho_0)$ going to zero then that choice is infeasible. Otherwise (k, ρ_0) may be feasible. We can thereby characterize all (k, ρ_0) pairs that may be feasible. The largest value of $k\log(1 + \rho_0)$ taken over these pairs gives us an upper bound on the throughput.

Note that this approach is general and can be used for any pdf, but requires a calculation of

$$p_s = \mathrm{P}\left(\frac{P\gamma_1}{\sigma^2 + P\sum_{i=2}^{k}\gamma_i} \geq \rho_0\right)$$

where all the channel coefficients in the SINR expression are drawn i.i.d. according

to $f_n(\gamma)$. For certain densities, such as the exponential, we may compute p_s and get an upper bound as follows.

If $f_n(\gamma) = e^{-\gamma}$, then

$$p_s = \mathrm{P}\left(\frac{P\gamma_1}{\sigma^2 + P\sum_{i=2}^{k}\gamma_i} \geq \rho_0\right) = \frac{e^{-\frac{\sigma^2}{P}\rho_0}}{(1+\rho_0)^{k-1}}.$$

With this,

$$M_n(k,\rho_0) = \binom{n}{k}\binom{n-k}{k}k!\frac{e^{-\frac{\sigma^2}{P}k\rho_0}}{(1+\rho_0)^{k(k-1)}}.$$

We now want to characterize (k,ρ_0) pairs for which $M_n(k,\rho_0)$ does not go to zero. We have

$$\begin{aligned} M_n(k,\rho_0) = \frac{n!}{(n-2k)!k!}p_s^k \leq \frac{n!}{(n-2k)!}p_s^k &\leq n^{2k}\frac{e^{-\frac{\sigma^2}{P}k\rho_0}}{(1+\rho_0)^{k(k-1)}} \\ &\leq \left(n^2\frac{1}{(1+\rho_0)^k}\right)^k = e^{k(2\log n - k\log(1+\rho_0))}. \end{aligned}$$

If k goes to infinity (with n) and $2\log n - k\log(1+\rho_0)$ is negative then $M_n(k,\rho_0)$ goes to zero. Therefore, for k going to infinity, we have $k\log(1+\rho_0) \leq 2\log n$ as a bound on the throughput. If k is constant, it is easy to see that $1+\rho_0$ cannot grow faster than n^2, hence the throughput is again limited by $k\log n^2 = 2k\log n$ where k is now a constant. Thus we have shown an upper bound of $c\log n$ on the throughput. This coincides (to within a multiplicative constant) with the throughput obtained in our achievability result (Section 2.8.4). In our scheme it turns out that using two hops is optimal for any n. Hence, although the upper bound derived here is on $k\log(1+\rho_0)$, it matches the achievability result for $\frac{k}{h}\log(1+\rho_0)$ very closely.

2.10 Simulations

Theorem 2.1 gives a very specific achievability result but equation (2.4) involves a constant α that is not explicit. This constant has its origins in Theorem 2.4 where the number of vertex-disjoint paths is computed. When we are confronted with a

specific network with a finite number of nodes n, we would like an explicit estimate of the number of non-colliding paths. In this section we provide such an estimate; we also briefly introduce the notion of "bad" edges, discuss decentralized algorithms for attaining our achievability results, and provide computer simulations of some of the networks analyzed in Section 2.8.

2.10.1 Non-colliding paths

In Section 2.5 we use a result of [43] to establish the existence of non-colliding paths. In this section, we present a constructive method of obtaining these paths and analyze the expected number of non-colliding paths thereby obtained. The algorithm we present is used extensively in Section 2.10.2.

We begin by choosing nodes $1, \ldots, n/2$ as source nodes and nodes $n/2+1, \ldots, n$ as their respective destination nodes. For the first source-destination pair, a shortest path connecting them (using only links that exceed β) is found. This is done using a standard breadth-first search algorithm [44] in which a rooted tree is constructed. All of the nodes begin by being "undiscovered." The source node acts as the root of the tree (at depth zero) and is labeled as "discovered." We then find all the nodes that are its neighbors and call them discovered. These are at distance one from the source and hence at depth one in the breadth-first search tree. The nodes at depth one are then processed successively. All of the neighbors of each node that are still undiscovered are put in the tree at depth two and their labels are changed to discovered. The process continues till there are no undiscovered nodes. Clearly, each node appears at most once in the tree. A shortest path from the source (root) to the destination is obtained by simply finding that node in the tree and moving up the tree to the source node. If the destination does not appear in the tree it has no path to the source.

Once the shortest path for the ith source-destination pair is established it is recorded and all n nodes are relabeled as "undiscovered"; the entire process is repeated to find the shortest path for the $(i+1)$st source-destination pair. This is done till paths are found for all $n/2$ pairs.

We then eliminate colliding paths on this list, starting with the first source-destination pair. If a node used on the path between s_1 and d_1 collides with a node on some other path, we eliminate path 1, otherwise we keep it. We proceed in order and eliminate the ith path if it collides with any of paths $i+1, i+2, \ldots, n/2$ and keep it otherwise. Note that since we start with shortest paths, a relay never appears more than once on a particular path.

Let us bound the probability that paths i and j collide for $i \neq j$. Without loss of generality we can set $i = 1$ and $j = 2$. We now have

$$\begin{aligned}
&\mathrm{P}(\text{path 1 collides with path 2}) \\
= \;& \mathrm{P}\left((s_1 = r_{2,1}) \cup \bigcup_{j=1}^{h-1} (r_{1,j} = r_{2,j-1} \cup r_{1,j} = r_{2,j} \cup r_{1,j} = r_{2,j+1}) \cup (d_1 = r_{2,h-1})\right) \\
\leq \;& \mathrm{P}(s_1 = r_{2,1}) + \sum_{j=1}^{h-1} \mathrm{P}(r_{1,j} = r_{2,j-1}) + \sum_{j=1}^{h-1} \mathrm{P}(r_{1,j} = r_{2,j}) + \sum_{j=1}^{h-1} \mathrm{P}(r_{1,j} = r_{2,j+1}) + \mathrm{P}(d_1 = r_{2,h-1}) \\
= \;& \frac{3h-1}{n-2} \qquad (2.21)
\end{aligned}$$

The inequality is a standard union bound and the last equality is because the $h-1$ relay nodes on the ith path are drawn uniformly at random from the set of all nodes of the graph (excluding s_i and d_i). (We assume that the algorithm that chooses the shortest path for (s_i, d_i) does not use any knowledge of the previously chosen $i-1$ paths.)

Denote by D_i the event of keeping the ith path. This event comprises the intersection of the events that the ith path does not collide with the $(i+1)$st through $(n/2)$th paths. These $n/2 - i$ events are identical although they are not necessarily independent. However, for the purposes of an approximation we may *assume* they

are independent and compute $\mathrm{P}(D_i)$ as follows.

$$\begin{aligned}
\mathrm{P}(D_i) &\approx \prod_{j=i+1}^{n/2} \mathrm{P}(\text{paths } i \text{ and } j \text{ do not collide}) \\
&= (\mathrm{P}(\text{paths } i \text{ and } i+1 \text{ do not collide}))^{n/2-i} \\
&= (1 - \mathrm{P}(\text{paths } i \text{ and } i+1 \text{ collide}))^{n/2-i} \\
&= (1 - \mathrm{P}(\text{paths 1 and 2 collide}))^{n/2-i} \\
&\geq \left(1 - \frac{3h-1}{n-2}\right)^{n/2-i}.
\end{aligned}$$

The inequality is a consequence of (2.21). We expect the inequality to be an approximate equality when h is small. The expected number of successful paths is then

$$\begin{aligned}
\text{Expected \# non-colliding} &= \sum_{i=1}^{n/2} \mathrm{P}(D_i) \\
&\approx \sum_{i=1}^{n/2} \left(1 - \frac{3h-1}{n-2}\right)^{n/2-i} \\
&= \frac{n-2}{3h-1}\left(1 - \left(1 - \frac{3h-1}{n-2}\right)^{n/2}\right) \qquad (2.22) \\
&\approx \frac{n-2}{3h-1} \qquad (2.23)
\end{aligned}$$

because $(1 - x/n)^{n/2} \approx e^{-x/2}$ decreases rapidly with x. This calculation, although based on an incorrect independence assumption is often useful to get an estimate of the number of non-colliding paths that we can expect to find.

We observe that in [43] vertex-disjoint paths are found successively and the nodes that are used in paths for source-destination pairs $1, \ldots, i$ are eliminated entirely from the graph before finding the path for the $(i+1)$st pair. The paper adroitly proves that at each stage the remaining graph has edges that are "approximately" i.i.d. (from the appropriate distribution). The approximation we use above deals with the loss of the i.i.d. property by simply ignoring it. Figure 2.4 shows that the approximations (2.22) and (2.23) can be very accurate. The figure shows the number of computer-found non-colliding paths obtained in the shadow-fading model in Section 2.8.1 with

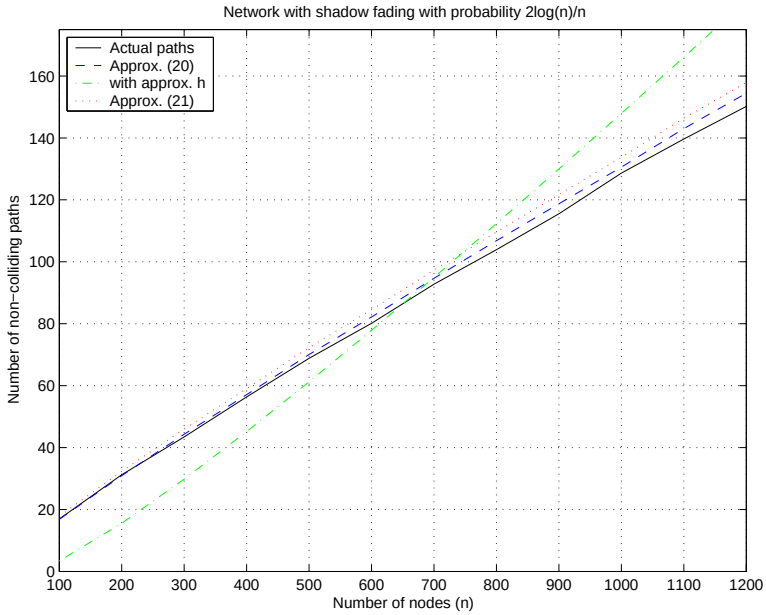


Figure 2.4: Number of computer-found non-colliding paths versus n for a shadow-fading model with connection probability $2(\log n)/n$ (solid curve) versus n. Also shown are the approximation (2.22) (dashed curve closest to solid curve) and the approximation (2.23) (next-closest dashed curve) using values of h obtained in the computer simulation. The dash-dotted curve is (2.22) computed using $h = \log(n)/\log(np)$.

link probability $p = 2(\log n)/n$. (We provide more details about this simulation in Section 2.10.2.) The most accurate approximation is obtained when the number of hops h in (2.22) and (2.23) is also taken from the simulation. However, we may always approximate the number of hops before the simulation as $h = \log(n)/\log(np)$. This final approximation is presented as the dash-dotted curve.

2.10.2 Simulations

We revisit some of the examples analyzed in Section 2.8 to see how well our analytical predictions match computer-generated simulations. We begin with the shadow-fading network analyzed in Section 2.8.1.

Figure 2.5 shows the aggregate throughput and minimum SINR of a shadow-fading network as a function of the number of nodes n in a computer-generated simulation where the channel connections are chosen as in Section 2.8.1. The analytical results

suggest that for best throughput we should choose $p = (\log n + \omega_n)/n$ for ω_n going to infinity arbitrarily slowly. We therefore choose $p = 2(\log n)/n$. The computer simulation begins by establishing a network of n connections whose channels are drawn i.i.d. according to (2.15). Non-colliding paths (using the method described in Section 2.10.1) are established and the minimal SINR obtained along the ith path, denoted $\rho_{0,i}$, is found. We are assured that an SINR of $\rho_{0,i}$ can be supported by the path and we use this rather than the threshold of ρ_0 that has been used in the analysis. Although the threshold of ρ_0 is significant in obtaining the analytical throughput guarantee, we believe that the notion of the minimum SINR along a path is more useful in a practical system. The quantity $\log(1 + \rho_{0,i})$ is then computed, weighted by the number of hops on path i, summed over i, and then normalized by the total number of hops contained in all paths. This gives a measure of the throughput per path, where paths that are longer (have more hops) count more heavily in the average. This throughput-per-path is then multiplied by the number of non-colliding paths and divided by the average number of hops to provide the aggregate throughput. Typically, we expect and observe only a small variation in the path lengths. Therefore, whether we divide by the average or largest path length does not make much difference.

The throughput shown in Figure 2.5 is an increasing function of n whose y-axis is labeled on the left. The minimal SINRs obtained along the ith path $\rho_{0,i}$ are averaged over i and displayed as a decreasing curve whose y-axis is labeled on the right. As predicted in Section 2.8.1, the aggregate throughput grows nearly linearly. We see that the average SINR per path, decreases slowly with n, especially when n is large; Section 2.8.1 shows that the SINR should go to zero as $1/\log\log n$.

The following applies to all simulations described in this section: (i) Computer simulations were repeated and averaged approximately 100–200 times, depending on the size of the network and variability of the results; (ii) The nodes have unit transmit power $P = 1$ and noise variance $\sigma^2 = 0.1$. Hence, on a unit channel and in the absence of interference, the SNR is 10 dB; (iii) We do not prescribe an SINR threshold. Rather, we accept any non-colliding path and use its resulting SINR

in our averages. We believe this to be reasonable in practice (the threshold ρ_0 is only the guaranteed minimum); (iv) The figures often show two plots; the aggregate throughput generally given by an increasing function of n and whose scale is on the left y-axis, and the average minimum SINR generally given by a decreasing function of n and whose scale is on the right y-axis; (v) Although the analysis uses logarithms with base e, the throughputs in the figures are given in bits/channel-use.

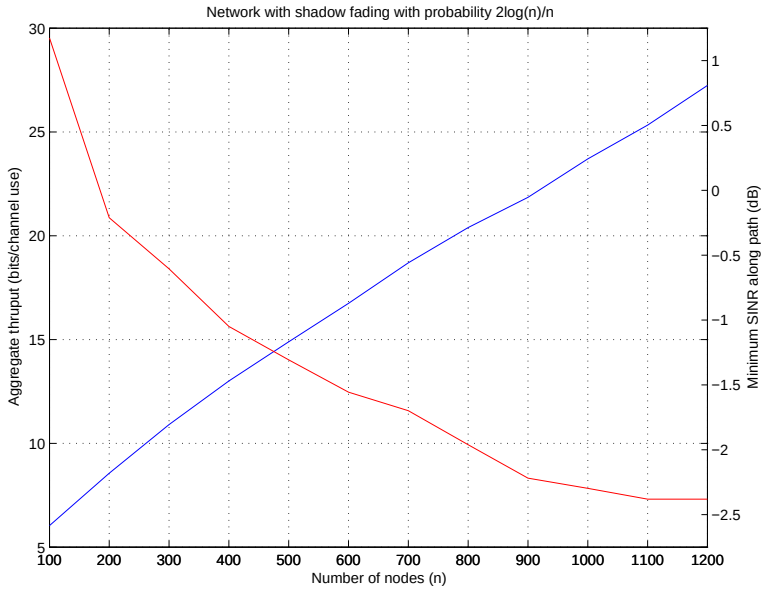


Figure 2.5: Aggregate throughput and minimum SINR versus number of nodes n in a shadow-fading network with connection probability $p = 2(\log n)/n$. The left y-axis contains the scale for this increasing function of n. We see that the aggregate throughput increases nearly linearly. The average SINR obtained along the paths (see scale on the right y-axis) drops with n, and according to the results in Section 2.8.1 should go to zero as $1/\log\log n$.

Figure 2.6 shows the aggregate throughput and minimum SINR of the same shadow-fading network, this time as a function of p for a fixed $n = 1000$ nodes. We see from the figure that the maximum throughput is attained when $p \approx 0.008$. Section 2.8.1 predicts that the maximum throughput is achieved when $p = (\log n + \omega_n)/n = 0.0069 + \omega_n/n$. Ignoring the ω_n term, we see a good match between the theory and the simulation.

Figure 2.7 shows the aggregate throughput and minimal SINR of a network with exponential fading analyzed in Section 2.8.4 as a function of n. For large enough n

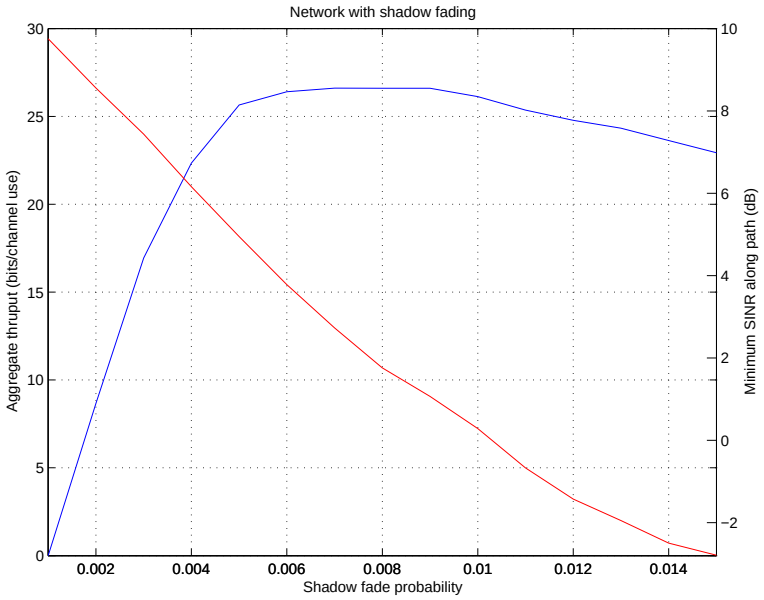


Figure 2.6: Aggregate throughput and minimum SINR versus connection probability p in a shadow-fading network of 1000 nodes. We see that the throughput is maximized at $p \approx 0.008$, which is not far from $(\log 1000)/1000 \approx 0.0069$, the large-$n$ maximizing p predicted in Section 2.8.1.

the optimum threshold is $\beta = (\log n)/2$ and k should be chosen as large as possible. For purposes of illustration, we therefore choose k as large as possible, even for the relatively small values of n that we consider. (In this particular example smaller values of k can yield higher total throughput when n is small.) The result is a throughput that grows approximately logarithmically with n, as predicted theoretically. The figure also shows that choosing a β that is constant has a detrimental effect on the throughput. Similarly, choosing a β that grows faster than logarithmically would also be detrimental.

Figure 2.8 shows the aggregate throughput and minimum SINR of the decay-density network (as a function of n) described in Section 2.8.2. The parameters used in the simulation are $d = 1$, $\Delta = 1$, and $m = 3$. This is equivalent to placing nodes with unit spacing in a two-dimensional lattice and assuming a power-decay that decreases as $1/r^3$. The figure shows that the throughput grows approximately linearly, as predicted by equation (2.20).

These simulations show that Theorem 2.1, although designed for large n, is also

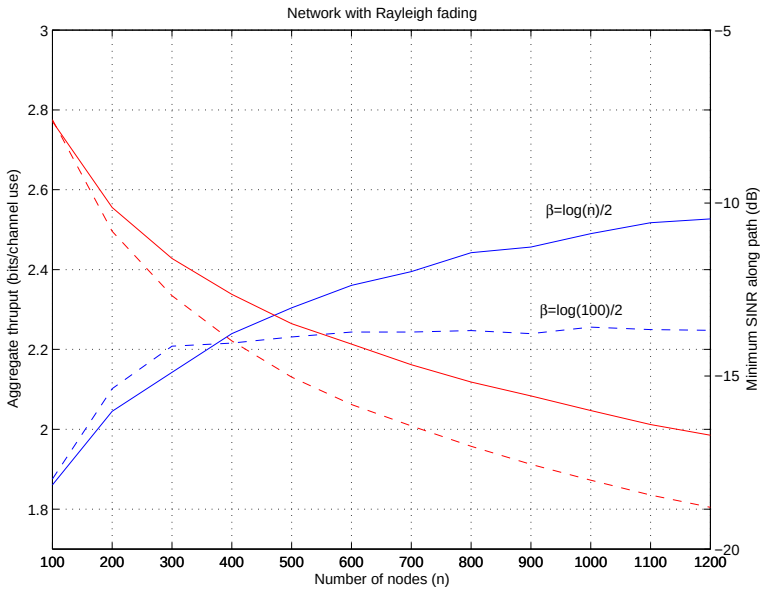


Figure 2.7: Aggregate throughput and minimum SINR versus number of nodes n in a network with exponential fading. We see that the throughput grows logarithmically using the optimum β computed in Section 2.8.4. The average SINR obtained along the paths decays approximately as $(\log n)/n$. Shown in dashed lines is the detrimental effect of choosing a constant $\beta = (\log 100)/2$.

an accurate predictor for finite n.

2.11 Conclusions

Our model for shared-medium wireless networks uses channels chosen according to a common distribution. We have devised a method of operating this network using relays and provided an achievable aggregate throughput as a function of the distribution. Distributions that have a certain sparsity of "good" connections seem to fare best and provide near-linear throughputs. We show that there exists an optimal amount of shadow-fading—any more or any less degrades the throughput. We hope that these results provide guidelines to the design of networks including, paradoxically, possible obstacle placement if the network is "over-connected."

We have given a brief description of an upper bound on the achievable throughput. In general, we do not know how sensitive our throughput results are to relaxing the i.i.d. assumption on the channel coefficients. The sensitivity to distance is low when

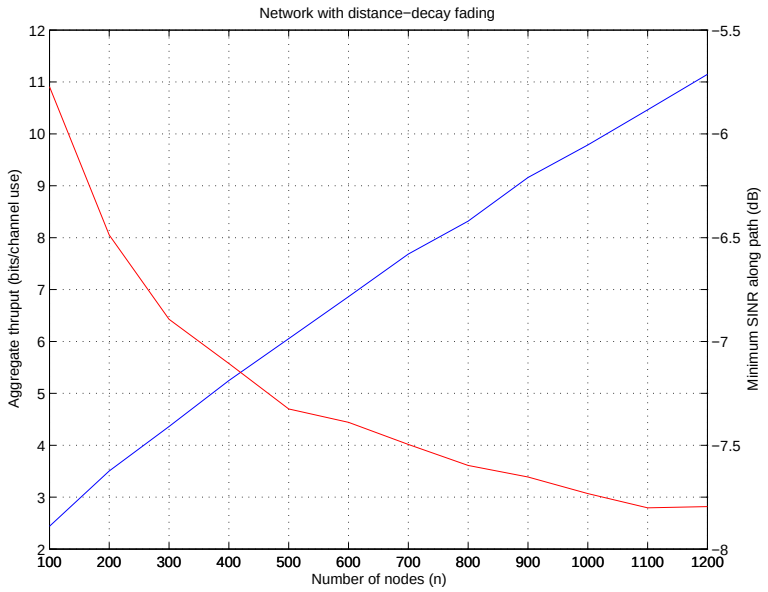


Figure 2.8: Aggregate throughput and minimum SINR versus number of nodes n in the decay-density network analyzed in Section 2.8.2. Equation (2.20) (for $m > 2$) predicts that the throughput should grow approximately linearly.

the channel coefficients are independent but have a distribution that depends on distance, as discussed in Section 2.8.

More practical issues remain to be addressed. For instance, the scheduling algorithm used in the simulations requires centralized knowledge of the channel connections. In a practical network we would expect sources and destinations to have knowledge of their own connections and determine suitable relaying paths. Ideally, we would then like the network to generate a non-colliding schedule in a decentralized manner. Another issue of interest is that of modeling a network that has both randomness and geometric distance-decay laws. One such model is proposed in [52] and presented in the next chapter.

Chapter 3

Two-Scale Models for Ad Hoc Networks

In the previous chapter, we proposed a network model in which connection strengths were independent of distance and were drawn independently and identically from some distribution. This model is very different from the purely geometric models like those of Kumar and Gupta, in which connection strengths depend entirely on the locations of nodes. In practice, we expect networks to have some local randomness properties in addition to the distance-decay effect that kicks in over longer distances. In this chapter we try to characterize such networks.

We propose a new model of wireless networks here, called "two-scale networks." At a local scale, characterized by nodes at most distance r apart, channel strengths are drawn independently and identically from a distance-independent distribution. At a global scale, characterized by nodes more than distance r apart, channel connections are governed by a Rayleigh distribution, with the power satisfying a distance-based decay law. Thus, random effects like obstacles and scatterers dominate local channel strengths while global scale channel strengths depend on distance.

For such networks, we propose a hybrid communication scheme, combining elements of [35] (for distance-dependent networks) and the previous chapter (for random networks [54]). For a particular class of two-scale networks with N nodes, we show that an aggregate throughput of the form $N^{\frac{1}{t-1}}/\log^2 N$ is achievable, where $t > 2$ is a parameter that depends on the distribution of the connection at the local scale and

is independent of the decay law that operates at a global scale. For $t < 3$, this offers a significant improvement over the $O(\sqrt{N})$ results of [35].

3.1 Introduction

As described in the introduction to the previous chapter, sensor and ad hoc networks have seen much research activity in recent times. The first major result of the field was by Kumar and Gupta [35] where a network of n nodes was studied. Strengths of the connections between two nodes were determined entirely by the distance between them and followed a deterministic power scaling law. With this model, it was shown that a throughput that scaled like $\sqrt{n}$ was the best possible. This implied that the throughput per user fell like $\frac{1}{\sqrt{n}}$ which was quite discouraging. Except when nodes were allowed to approach each other [30], similar scaling laws were shown to hold [34, 29, 33, 39, 36].

From the research on multiple antenna systems, we know that rich scattering environments, leading to independent channel coefficients between transmit and receive antennas help achieve capacity linear in the number of antennas [28, 37]. Taking a cue from this, a network model with random connections was proposed in [51, 54]. (This is presented in the previous chapter.) In this, the channel strengths are independent of distance and geometry and are instead drawn identically and independently (i.i.d.) from a probability distribution function (pdf). This model is suitable for networks over a small area, where multipath and physical obstructions dominate and the decay laws associated with far-field effects do not kick in.

While the throughput that was possible with this model depended very strongly on the distribution that the channel strengths were drawn from, several distributions, including the Bernoulli and some heavy-tailed distributions led to throughputs that were almost linear in n. Thus the introduction of randomness changed the behavior of the system significantly.

In practice, we expect neither the deterministic model of [35] nor the random model of the previous chapter, [54], to hold. A combination of distance-dependent

connections and random connections would perhaps make for a better model. In this chapter, we propose and analyze such a model. We assume that N nodes are randomly and uniformly distributed on a sphere of radius R. Nodes that are within a distance r from each other are connected by channels that are distance-independent. These channel strengths are assumed to be drawn i.i.d. from a distribution, $f(\cdot)$. For nodes that are further apart than r, the channel connections obey a Rayleigh distribution with a mean power that depends on the distance between them and follows a distance-decay law, say $g(\cdot)$.

Such a model incorporates the far-field effects at a global level through the decay law, but also recognizes that obstructions play a role at a local scale. Furthermore, appropriate choices of r and R can help model a full scale of networks, from the purely geometric ones of [35] to the purely random ones of the previous chapter. A precise description of the model and the problem statement is in Section 3.2. Sections 3.3 and 3.4 study the scheduling and error-free communication properties of this model and the main result is stated in Section 3.5. Examples and conclusions are presented at the end. Not surprisingly, a combination of the techniques found in [35] and [54] are employed throughout this chapter.

3.2 Network Model

Consider a network with N nodes that are uniformly and randomly distributed on the surface of a sphere of radius R. We use a sphere rather than a planar disk to separate edge effects and have symmetry between all nodes. Also, the standard convention of measuring distances along great circles will be followed.

The channel between nodes i and j is denoted by $h_{i,j} = h_{j,i}$. Define the channel strength to be $\gamma_{i,j} = |h_{i,j}|^2$. The average channel strength is assumed to be distance-dependent for nodes that are more than a certain distance, say r, apart and independent of distance for nodes that are within a distance r.

More precisely, for nodes that are within a distance r, the channel strengths are drawn i.i.d., according to a p.d.f., say $f(\gamma)$. Let the expected value corresponding to

this be denoted by μ_γ.

If nodes i and j are at a distance $l(i,j) > r$ from each other, we model $h_{i,j}$ to be a Rayleigh distributed random variable with its power (or second moment), $E|h_{i,j}|^2$, given by $cg(l(i,j))$ where $g(x)$ is used to model the distance-dependence and c is a constant. This gives us that the corresponding $\gamma_{i,j}$ is drawn from an exponential distribution with $cg(x)$ as its mean, i.e., $cg(x)\exp(-\gamma/cg(x))$. Typically, $g(x)$ is a decreasing function such as $\frac{1}{x^m}$ with $m > 2$ or $\frac{e^{-\delta x}}{x^m}$ and c is chosen such that $cg(r)$ equals μ_γ. This is done to ensure that the expected value of $\gamma_{i,j}$ does not change abruptly as the distance between i and j changes from being less than r to being greater than r. Therefore, $c = \frac{\mu_\gamma}{g(r)}$.

Denote by $p_x(\gamma)$ the distribution from which the channel strength between two nodes with distance x between them is drawn. Then we have

$$p_x(\gamma) = \begin{cases} f(\gamma) & \text{if } x \leq r \\ \frac{\mu_\gamma g(x)}{g(r)} \exp(-\gamma \frac{g(r)}{\mu_\gamma g(x)}) & \text{if } x > r \end{cases}.$$

3.2.1 Successful Communication

The concept of successful communication is identical to that stated in the previous chapter. We repeat it here for completeness. Assume that node i wishes to transmit signal x_i. We assume that x_i is a complex Gaussian random process with zero mean and unit variance. Each node is permitted a maximum power of P watts.

We incorporate interference and additive noise in our model as follows. Assume that l nodes $i_1, i_2, \ldots, i_l$ are simultaneously transmitting signals $x_{i_1}, x_{i_2}, \ldots, x_{i_l}$ respectively. Suppose that node j is the intended receiver of the signal x_{i_1}. Then, the signal received by node $j(\neq i_1, \ldots, i_l)$ is given by

$$y_j = \sum_{t=1}^{l} \sqrt{P} h_{i_t,j} x_{i_t} + w_j \tag{3.1}$$

where w_j represents additive noise. The additive noise variables $w_1, \ldots, w_N$ are i.i.d., drawn from a complex Gaussian distribution of zero mean and variance σ^2 ($w_i \sim$

$\mathcal{CN}(0, \sigma^2)$). The noise is statistically independent of x_i.

In equation (3.1), assume that only node i_1 wishes to communicate with node j and the signals $x_{i_2}, \ldots, x_{i_l}$ are interference. Then the signal-to-interference-plus-noise ratio (SINR) for node j is given by

$$\rho_j = \frac{P\gamma_{i_1,j}}{\sigma^2 + P\sum_{t=2}^{l} \gamma_{i_t,j}}.$$

Note that some of the interference terms will come from the exponential distribution and the others will be drawn from $f(\gamma)$, depending upon the distance of the interferer from j. We assume that transmission is successful when the SINR exceeds some ρ_0. If the SINR is less than ρ_0, we will say that an error has been made.

3.2.2 Network Operation and Throughput

We suppose that K nodes $s_1, \ldots s_K$ are randomly chosen as sources. For every s_i, a destination node, say d_i, is chosen at random, thus making K source-destination pairs. We assume that these $2K$ nodes are all distinct and therefore $K \leq N/2$. Source s_i wishes to transmit message W_i to destination d_i and has encoded it as signal x_i.

Communications are assumed to occur using a series of hops. Every source-destination pair (s_i, d_i) uses a sequence of relay nodes, each of which are expected to decode the message x_i and retransmit it in the next time slot, using power P. We expect several messages to be making hops simultaneously and therefore the relay nodes have to decode in the presence of interference. With this in mind, we impose the constraint that no relay node be asked to decode two messages simultaneously. We also assume that no relay node can receive and transmit in the same time slot. These properties will define a *non-colliding* schedule of relaying.

Assume that all K messages reach the intended destinations in (at most) H time slots. Assume that a fraction ϵ of messages fail to reach the intended destination due to decoding or scheduling errors. Each message contains at least $\log(1 + \rho_0)$ bits of

information since ρ_0 is the SINR threshold. Therefore, we define the throughput as

$$T = (1-\epsilon)\frac{K}{H}\log(1+\rho_0) \tag{3.2}$$

Note that all the quantities above may depend on N. Typically, we force ϵ to go to zero. In the rest of this chapter, we present a scheme of scheduling and communicating and analyze the throughput as well as performance of this scheme. Our concern will primarily be with arbitrarily large values of N. Thus, we will obtain an asymptotic achievability result for the throughput T.

3.3 Relaying Scheme

In this section we determine the scheduling of the relay nodes for the multihop protocol. We do this through various constructions, including Voronoi tessellations, a superschedule and many subschedules. We will borrow techniques from [35] and the previous chapter ([54]), and put them together in a suitable manner to perform scheduling for the proposed hybrid model.

3.3.1 Tessellations and Cell-aggregates

Recall the concept of a Voronoi tessellation, used extensively in [35]. Lemma 4.1 of [35] establishes the existence of a Voronoi tessellation of the surface of the unit sphere where each Voronoi cell contains a disk of radius δ and is contained in a disk of radius 2δ for any $\delta > 0$. We will use this result for the surface of the sphere of radius R. (This can be done by using the original result for δ/R rather than δ and then scaling the obtained tessellation by a factor of R.) Denote by $\mathcal{T}(x)$ a tessellation of the surface of the sphere of radius R where each Voronoi cell contains a disk of radius x and is contained in a disc of radius $2x$. In particular, consider a tessellation $\mathcal{T}(r/12)$ where r is the radius within which channel strengths are distance-independent and are drawn i.i.d. from $f(\gamma)$. Cells of this tessellation will be labeled S_i. It is easy to show that in such a tessellation, for any cell, S_i, it and all its neighboring cells

are contained in a disk of diameter r. (A similar, though slightly different, result is shown in Lemma 4.2 of [35].) Thus, every connection within this set of cells is drawn i.i.d. according to $f(\gamma)$. Recall that the area of a circle of radius x on the surface of a sphere of radius R is given by $A(x) = 4\pi R^2 \sin^2 \frac{x}{2R}$. Using this fact, it is possible to show that the number of cells that are neighbors of a given cell is bounded by a constant, say c_1.

3.3.2 Determining a Superschedule

Assume that such a tessellation of the surface of the sphere is done once and fixed. We refer to this as $\mathcal{T}_0(r/12)$. Every node belongs to some S_i. (Nodes lying on cell boundaries are assigned arbitrarily.) Consider the source-destination pair (s_i, d_i). Denote by L_i the line segment connecting them. This segment passes through several cells in order as it traverses from s_i to d_i. Note that the maximum number of cells it can pass through is $M = c_2 \frac{R}{r}$ for some constant c_2. Denote these cells, in sequence, by $s_i \in S_{i,0}, S_{i,1}, S_{i,2}, \ldots, S_{i,M} \ni d_i$. (Some sequences may, in actuality, be shorter than M.) We will refer to the set of cells $S_{1,t}, S_{2,t}, \ldots, S_{K,t}$ as the tth *layer* of cells.

The schedule described above tells us the cells that each message must pass through in a particular layer. We now decide which node in each cell is responsible for each message in each layer of transmission.

There are at least $\frac{4\pi R^2}{A(r/6)} = 1/\sin^2 \frac{r}{12R}$ cells in $\mathcal{T}_0(r/12)$. The K sources are assumed to be uniformly distributed on the surface of the sphere. Therefore, each cell has at most $K \sin^2 \frac{r}{12R} = k_1$ sources. (This can be made more rigorous.) Thus, that cell occurs in the zeroth layer around k_1 times. In general, a cell occurs in the tth layer around k_1 times.

We require that the K nodes that act as relay nodes in one layer be distinct from each other as well as the K nodes occurring in the previous layer. This is equivalent to requiring the k_1 relay nodes in each cell of the tth layer to be distinct from each other as well as the k_1 nodes from the same cell occurring in the previous layer. In the zeroth layer of transmission, this condition is trivially met since the K distinct

original source nodes (around k_1 of them occurring in each cell) start out having the messages that need to be relayed and there is no previous layer. We wish to have such distinct nodes for the ith layer assuming that such nodes for each layer up to the $(i-1)$th have already been determined. Let us determine the conditions under which this is possible.

Consider a specific cell in $\mathcal{T}_0(r/12)$. This is expected to have k_1 distinct nodes that are the chosen relays in the $(i-1)$-th layer. This cell also occurs k_1 times in the i-th layer and we wish to assign a further k_1 distinct relay nodes for each occurrence. The total number of nodes in this cell is at least $N/(\text{maximal number of cells}) = N/(4\pi R^2/A(r/12)) = N\sin^2\frac{r}{24R} = n_1$. Therefore our condition of distinct nodes can be met if $2k_1 \leq n_1$. After simplification, this gives

$$K \leq N/(8\cos^2\frac{r}{24R}).$$

Once this condition is satisfied, we can assign a distinct relay node for each of the K messages in each layer. The relay node in layer t that is responsible for message i will be called $s_{i,t}$. (Clearly, $s_{i,t} \in S_{i,t}$.) We refer to the K sequences $s_i = s_{i,0}, s_{i,1}, \ldots, s_{i,M} = d_i$ for $i = 1, \ldots, K$ as the superschedule.

3.3.3 Non-colliding Subschedules

It now remains to decide how to route the message i from its relay node in layer t, namely $s_{i,t}$, to its relay node in layer $(t+1)$, namely $s_{i,t+1}$. We refer to this as subscheduling. We consider time slots in blocks of size h, where h denotes the (maximal) number of hops required for a message to be transmitted from $s_{i,j}$ to $s_{i,j+1}$. In a specific block of time slots, say from $vh+1$ to $(v+1)h$, some constant fraction c_3 of all cells are chosen at random and called active cells. We choose $c_3 < 1/c_1$ where c_1 is an upper bound on the number of neighbors of a cell. Denote the set of chosen cells by T_v. Consider the cells that are not in T_v. Let j be such a cell. If one of the neighbors of j is in T_v, assign j to it. If more than one of the neighbors of j are in T_v, this assignment can be done randomly. Thus, for each of the $|T_v|$ originally chosen

cells, we now have $|T_v|$ *cell-aggregates* that are active. (Some of these may consist of just one cell, namely, the originally chosen cell.) Figure 3.1 demonstrates this. In the v-th block of time slots, communication occurs only within the T_v cell aggregates and not across one aggregate to another. Since any cell and its neighbors can be put inside a circle of diameter r, connections with an aggregate are drawn i.i.d. from $f(\gamma)$. We make use of this fact in determining h and a non-colliding subschedule in Lemma 3.1.

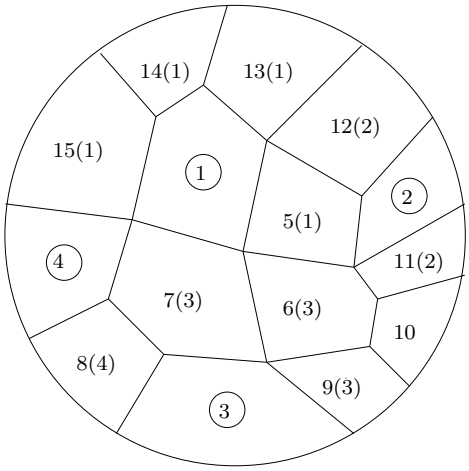


Figure 3.1: Cells 1, 2, 3, 4 (circled) are originally chosen to be in T_v. The remaining cells are then assigned as indicated in parentheses. For example, 13 gets assigned to 1 and 6 to 3. Cell 10 remains unassigned. The aggregate corresponding to cell 3 consists of cells 3, 6, 7, and 9.

A particular choice of T_v leads to some pairs of adjacent cells not being in the same cell-aggregate. For a pair that gets split into two cell-aggregates, the relays in one cell that have the next relay in the other cell are unable to communicate with each other in the v-th block of time slots. However, there is a probability that in another set, say T_w, this pair does not get split up. Let B be the number of sets we have to choose in order for every pair of adjacent cells to have been chosen in the same aggregate at least once.

Let i and j be adjacent cells. They can be in the same cell-aggregate in a randomly obtained T_v if ($i \in T_v$, $j \notin T_v$ and j gets assigned to i) or vice versa. By symmetry,

both cases are equally likely. Therefore,

$$\begin{aligned}
&\mathrm{P}(i,j \text{ are in the same cell-aggregate}) \\
&= 2\ \mathrm{P}(i \in T_v, j \notin T_v, j \text{ gets assigned to } i) \\
&= 2\ \mathrm{P}(i \in T_v)\mathrm{P}(j \notin T_v | i \in T_v)\mathrm{P}(j \text{ is assigned to } i | i \in T_v, j \notin T_v) \\
&\geq 2\ c_3(1-c_3)\frac{1}{c_1}.
\end{aligned}$$

The last expression follows since a fraction c_3 of cells is chosen at random to be in T_v. Therefore i is in T_v with probability c_3 and j is not in T_v with probability $(1-c_3)$ independently of i. Finally, j has at most c_1 neighbors, including i. If some x of them are chosen in T_v (and i is one of them), the probability of j being assigned to i is $1/x \geq 1/c_1$.

Let $c_4 = 2c_3(1-c_3)\frac{1}{c_1}$. Any choice of $c_3 < c_1/2$ ensures that $c_4 < 1$. Therefore, the probability that i and j are not in the same cell-aggregate in B choices for sets of cell-aggregates is bounded above by $(1-c_4)^B = e^{B\log(1-c_4)}$. If we choose B to be $\log N$, this behaves as $N^{\log(1-c_4)}$ which goes to zero as N goes to infinity. (It is clear that B can be chosen to be any function that goes to infinity for large N.)

Consider a block in which a particular cell-aggregate is active. Assume that it consists of $c_5 \leq 1 + c_1$ cells. Each cell has around k_1 relays that wish to transmit and k_1 relays that wish to receive in a particular layer. Thus, we expect there to be no more than $c_5 k_1$ transmissions that need to take place while that cell-aggregate is active. We denote the actual number of transmissions by k. In addition, the cell-aggregate lies entirely in a circle of diameter r; therefore all the connection strengths within it are drawn i.i.d. from the distribution $f(\gamma)$. Let $n = c_5 n_1$ be the total number of nodes in the aggregate.

In this subnetwork of n nodes with i.i.d. connections we seek a schedule of k non-colliding paths from the set of transmitting relays to the set of receiving relays. This is exactly the problem that is addressed in the previous chapter, or [54].

3.3.4 Good Edges and Vertex-Disjoint Paths

We reproduce the solution presented in the previous chapter. The channels that are stronger than a chosen parameter β are called *good*. All communications take place over good channels. Since channels are drawn i.i.d. from $f(\gamma)$, for every channel, there is a probability $p = P(\gamma \geq \beta)$ of its being good. We now construct a graph on n vertices where each vertex represents a node of the network. An edge is drawn between two vertices if the channel between the corresponding nodes is good. Thus, we obtain a graph on n vertices where edges are drawn i.i.d. from a Bernoulli distribution of parameter p.

Such a graph fits a standard random graph model called $\mathcal{G}(n, p)$. This model is well-studied and we appeal to an existing result in the literature to help us with our scheduling. We seek k non-colliding paths that go from the set of tth layer relay nodes to the respective $(t+1)$th layer relay nodes. In [43], an identical problem is studied, but the condition on the paths is stricter still – no two paths can share a vertex. In other words, the paths must be vertex-disjoint. We state here the result of [43] as it applies to our problem.

Lemma 3.1. *Suppose that $G = G(n, p)$ and $p \geq \frac{\log n + \omega_n}{n}$, where $\omega_n \to \infty$. Then there exists a constant $\alpha > 0$ such that, with probability approaching one, there are vertex-disjoint paths connecting x_i to y_i for any set of disjoint, randomly chosen node pairs*

$$F = \{(x_i, y_i) | x_i, y_i \in \{1, \ldots, n\}, i = 1, \ldots, k\}$$

provided $k = |F|$ is not greater than $\alpha n \frac{\log np}{\log n}$.

The x_is of the result above are the transmitting relays (from the tth layer) and the y_is are the corresponding receiving relays (from the $(t+1)$th layer). From Section 3.3.2, we know that these are all distinct nodes. We have $k = c_5 k_1 = c_5 K \sin^2 \frac{r}{12R}$ and $n = c_5 N \sin^2 \frac{r}{24R}$. Therefore the above theorem establishes the existence of the

required non-colliding paths if $c_5 K \sin^2 \frac{r}{12R} \leq \alpha c_5 \left(N \sin^2 \frac{r}{24R}\right) \frac{\log c_5 N \sin^2 \frac{r}{24R} p}{\log c_5 N \sin^2 \frac{r}{24R}}$ or

$$K \leq \alpha N \frac{\log c_5 N \sin^2 \frac{r}{24R} p}{\log c_5 N \sin^2 \frac{r}{24R}} \frac{1}{\cos^2 \frac{r}{24R}}.$$

Recall that for every block of h time slots, we have certain active cell-aggregates. Each time a cell-aggregate is active, we can appeal to the above theorem to get a satisfactory schedule. Additionally, it is possible to show that the lengths of the vertex-disjoint paths grow no faster than $\frac{\log n}{\alpha \log np}$. Therefore, the time slots required, h, are bounded above by $h \leq \frac{\log n}{\alpha \log np} = \frac{\log c_4 N \sin^2 \frac{r}{24R}}{\alpha \log c_4 N \sin^2 \frac{r}{24R} p}$.

Putting the results of this section together, we have the following result.

Theorem 3.2. *All K communications can be scheduled in $H = hMB = \frac{\log n}{\alpha \log np} \cdot c_2 \frac{R}{r} \cdot \log N$ time slots using non-colliding paths of length $hM = \frac{\log n}{\alpha \log np} \cdot c_2 \frac{R}{r}$ provided the following conditions hold.*

1. $K \leq N/(8 \cos^2 \frac{r}{24R})$.
2. $K \leq \alpha N \log np/(\log n \cdot \cos^2 \frac{r}{24R})$.

Here, $\frac{\log n + \omega_n}{n} \leq p \leq 1$ is a probability, $n = c_5 N \sin^2 \frac{r}{24R}$, α and c_5 are constants, and ω_n can be any function that goes to infinity.

Thus, the hybrid model allows us to schedule non-colliding paths using a combination of ideas from the deterministic model of [35] and the random model of the previous chapter. The next question to investigate is that of an appropriate SINR threshold, ρ_0 that determines the rate of the transmissions.

3.4 Probability of Error

All transmissions take place in the presence of noise and interference. The SINR threshold ρ_0 has to be carefully set so that it is not too low, but is low enough to ensure that most communications are successful. Let us investigate the SINR at any particular hop. Let us assume that node a is transmitting to node b. The power of

the transmission is P. All communications take place on channels that are good; that is, where $\gamma \geq \beta$. Therefore, the signal power is at least $P\beta$. The additive noise power is σ^2. There is interference from all other transmissions that occur in the same time slot. Some of these transmitting nodes lie within a distance r of the receiving node b and others lie farther.

Consider the interferers lying within a distance r. There are around $k_2 = K\frac{A(r/2)}{4\pi R^2} = K\sin^2\frac{r}{4R}$ of them, say $u_1, \ldots, u_{k_2}$ and the interference from them is given by

$$I_{\text{inside}} = P\sum_{i=1}^{k_2} \gamma_{u_i,b}.$$

The expected value of this is easily calculated and $EI_{\text{inside}} = Pk_2\mu_\gamma = PK\sin^2\frac{r}{2R}\mu_\gamma$.

The other interferers lie farther than a distance r from the b. Let us assume that there are K such interferers. (This in an overestimate since we have K paths in total and do not expect them all to be active at the same time.) Therefore, the total interference from them is given by

$$I_{\text{outside}} = P\sum_{i=1}^{K} \gamma_{u_i,b}$$

where the $\gamma_{u_i,b}$ are exponential random variables with mean $\frac{\mu_\gamma g(l(u_i,b))}{g(r)}$ and $l(u_i, b)$ is the distance between u_i and b.

We now calculate the expected value of I_{outside}. Let us represent the density of these interferers by $\kappa = K/4\pi R^2$. Consider an infinitesimally thin annulus of radius $t > r$ and width dt centered at b. Since we are on the sphere, the area of this annulus is less than $2\pi t dt$ and the number of interferers in this annulus is $\kappa 2\pi t dt$. In the expression for I_{outside} above, there are around these many terms with mean $\frac{\mu_\gamma g(t)}{g(r)}$. Therefore we have

$$EI_{\text{outside}} \leq \int_r^\infty P\kappa 2\pi t\frac{\mu_\gamma g(t)}{g(r)}dt = PK\frac{r^2\mu_\gamma}{2R^2}$$

in the case where $g(t) = 1/t^m, m > 2$.

We have this bound on the SINR for node b.

$$\rho_b \geq \frac{P\beta}{\sigma^2 + I_{\text{inside}} + I_{\text{outside}}}.$$

The probability that the SINR falls below some threshold ρ_0 is bounded as follows.

$$\begin{aligned} \mathrm{P}(\rho_b \leq \rho_0) &\leq \mathrm{P}\left(\frac{P\beta}{\sigma^2 + I_{\text{inside}} + I_{\text{outside}}} \leq \rho_0\right) \\ &= \mathrm{P}\left(I_{\text{inside}} + I_{\text{outside}} \geq \frac{P\beta}{\rho_0} - \sigma^2\right) \\ &\leq \frac{E(I_{\text{inside}} + I_{\text{outside}})}{\frac{P\beta}{\rho_0} - \sigma^2} \\ &\leq \frac{K \sin^2 \frac{r}{4R} \mu_\gamma + K \frac{r^2 \mu_\gamma}{2R^2}}{\frac{\beta}{\rho_0} - \frac{\sigma^2}{P}} \end{aligned} \tag{3.3}$$

where the Markov inequality and the expected values of the interferences have been used in the last line.

We will set the SINR threshold to

$$\rho_0 = \frac{P\beta}{\sigma^2 + a(Pk\mu_\gamma + PK\frac{r^2\mu_\gamma}{2R^2})} \tag{3.4}$$

where $a \geq 1$ can be suitably chosen to make transmissions error-free. This value of ρ_0 is chosen keeping in mind that the interference terms are expected to behave like their expected values for large networks. We use a to keep the threshold conservative.

Finally, we know that every message makes $hM = \frac{\log n}{\alpha \log np} c_2 \frac{R}{r}$ hops as described in Section 3.3. At each hop, the probability that the SINR falls below the threshold ρ_0 is as calculated above. With a simple union bound, similar to equation 2.7 of the previous chapter, it is possible to show that a message fails to reach its destination with probability ϵ where

$$\epsilon \leq \#\text{ hops} \cdot P(\rho_b \leq \rho_0) \leq \frac{\log n}{\alpha \log np} c_2 \frac{R}{r} \frac{1}{a} \tag{3.5}$$

The value of ρ_0 as given in (3.4) and the expression of (3.3) have been used.

3.5 Deriving the Main Result

We now have all the pieces we need to obtain the final result. Section 3.3 tells us the conditions for the existence of a non-colliding schedule and Section 3.4 tells us the conditions for communications to be successful with this schedule. We thus have the following result.

Theorem 3.3. *Consider a network of N nodes, uniformly and randomly distributed over the surface of a sphere of radius R. For two nodes within a distance r, channel strengths are drawn i.i.d. from a pdf $f(\gamma)$ with mean μ_γ. Otherwise they are drawn from an exponential distribution with a mean of $\mu_\gamma r^m/x^m$, where $x > r$ is the distance between them. Let $F(\gamma)$ denote the cumulative distribution function of $f(\gamma)$ and $Q(\gamma) = 1 - F(\gamma)$. Let $n = c_5 N \sin^2 \frac{r}{24R}$ where c_5 is a known constant. Choose any β such that $p = Q(\beta) = \frac{\log n + \omega_n}{n}$, where $\omega_n \to \infty$ as $n \to \infty$. Let $A(x) = 4\pi R^2 \sin^2 \frac{x}{2R}$. Then a throughput of*

$$T = (1-\epsilon)\frac{\alpha K r \log np \cdot \log\left(1 + \frac{P\beta}{\sigma^2 + a(PK\sin^2\frac{r}{4R}\mu_\gamma + PK\frac{r^2\mu_\gamma}{2R^2})}\right)}{\log n \cdot c_2 R \cdot \log N}$$

is achievable where α and c_2 are constants and K and $a \geq 1$ are chosen such that the following conditions are satisfied.

1. $K \leq N/(8\cos^2 \frac{r}{24R})$.
2. $K \leq \alpha N \log np/(\log n \cdot \cos^2 \frac{r}{24R})$.
3. $\epsilon \leq \frac{\log n}{\alpha \log np} \cdot \frac{R}{r} \cdot \frac{1}{a} \to 0$.

Proof. From Theorem 3.2 we know the number of hops required for a non-colliding schedule and the necessary conditions. From (3.4) and (3.5) in Section 3.4 we know ρ_0 and the condition for successful communications. Recalling that the throughput is $T = (1-\epsilon)\frac{K}{H}\log(1+\rho_0)$, we get the above theorem. □

Example 3.5.1. *Consider $f(\gamma) = \frac{1}{(1+\gamma)^t}$ with $t > 2$ as the distribution from which the channel strengths are drawn i.i.d. for nodes within a distance r from each other.*

We need $t > 2$ for μ_γ to be finite. We will assume that the other connections are exponential with the mean following a distance decay law of $g(x) = 1/x^m$ for $m > 2$. Choosing $p = \frac{2\log n}{n}$, we get a β that behaves like $(n(t-1)/2\log n)^{\frac{1}{t-1}} - 1$. Since r and R are fixed, we can approximate $\sin^2 \frac{r}{4R}$ with $c_6 \frac{r^2}{R^2}$. Therefore, $n = c_5 c_6 N \frac{r^2}{R^2} = c_7 N$ and $\beta \approx N^{\frac{1}{t-1}} / \log^{\frac{1}{t-1}} N$. One can choose K to be of the form $N/\log N$ and a of the form $\log\log N \frac{\log n}{\alpha \log np} \frac{R}{r}$. This satisfies the required conditions of the theorem and we get a throughput of $T = N^{\frac{1}{t-1}} / \log^2 N$. For t just greater than 2, this is almost linear but for $t > 3$, it falls below $\sqrt{N}$. It is interesting to note that m plays no role in this analysis.

3.6 Conclusions

We have proposed a two-scale network model in which local connections are drawn at random and global connections depend on a distance-based decay law. We have analyzed the throughput for this network and found that depending on the chosen parameters it can give a wide range of throughputs. Further generalization of this model is also possible. For instance, we can think of a three-scale model in which connections between nodes are i.i.d. within a distance r_1, become distance-dependent beyond a distance r_2 and are either distance-dependent or random for distances between r_1 and r_2. For this model, the connection strengths, γ, would be described by

$$p_x(\gamma) = \begin{cases} f(\gamma) & \text{if } x \le r_1 \\ \frac{r_2 - x}{r_2 - r_1} f(\gamma) + \frac{x - r_1}{r_2 - r_1} \frac{\mu_\gamma r_2^m}{x^m} \exp(-\gamma \frac{x^m}{\mu_\gamma r_2^m}) & \text{if } r_1 < x \le r_2 \\ \frac{\mu_\gamma r_2^m}{x^m} \exp(-\gamma \frac{x^m}{\mu_\gamma r_2^m}) & \text{if } x > r_2 \end{cases}. \tag{3.6}$$

Approaches similar to those used for two-scale models are expected to be useful for this model as well. We can also think of a mixture model, in which connection strengths change gradually from being i.i.d. to being distance-dependent. The connection

strengths for these are described by

$$p_x(\gamma) = \frac{R-x}{R} f(\gamma) + \frac{x}{R} \frac{1}{x^m} \exp(-\gamma x^m). \tag{3.7}$$

We see that the probability of i.i.d. connections decreases gradually (linearly) with distance and that of distance-dependent connections increases simultaneously. Developing approaches for the analysis of these networks is an interesting line of future work.

Chapter 4

Capacity of Wireless Erasure Networks

In the previous two chapters, we studied a network in which connections between nodes were drawn at random from a particular distribution or were dependent on distance. For such networks, we presented a throughput result that held for asymptotically large networks. In this chapter, we take a different view of the matter by introducing a class of networks called wireless erasure networks. These networks have a fixed number of nodes and each node is connected to a set of nodes by possibly correlated erasure channels. The network model incorporates the broadcast nature of the wireless environment by requiring each node to send the same signal on all outgoing channels. However, we assume there is no interference in reception. Such models are therefore appropriate for wireless networks where all information transmission is packetized and where some mechanism for interference-avoidance is already built in. We study multicast problems for these networks. We obtain the capacity under the assumption that erasure locations on all the links of the network are provided to the destinations.

This result has a very different flavor from the result of the previous chapter, in that it gives capacity for any particular network, for any number of nodes and any precise topology. This capacity is not asymptotic or inexact. The expression for the capacity takes into account the topology of the network and can be written for networks with any number of nodes. The restriction is that the network has to consist

of erasure links, with broadcast and no interference. Thus, the model is more specific than the random network model of the previous chapter and also gives a more precise result. We thus see that having a more restrictive, and thus simpler, model gives us tighter results, while more general models, like that of the previous chapter, force us to take a more approximate approach and perform an asymptotic analysis.

Coming back to the wireless erasure networks studied in this chapter, it turns out that the capacity region has a nice max-flow min-cut interpretation. The definition of cut-capacity in these networks incorporates the broadcast property of the wireless medium. We also show that linear coding suffices to achieve the capacity region.

4.1 Introduction

Determining the capacity region for general multi-terminal networks has been a long-standing open problem. An outer bound for the capacity region is proposed in [71]. This outer bound has a nice min-cut interpretation: The rate of flow of information across any cut (a cut is a partition of the network into two parts) is at most the corresponding cut-capacity. The cut-capacity is defined as the maximal rate that could be achieved if the nodes on each side of the cut could fully cooperate and also use their inputs as side-information.

This outer bound is not necessarily tight, in general. For instance, for the single relay channels introduced in [3], no known scheme achieves the min-cut outer bound of [71].

However, the max-flow min-cut outer bound is tight for *wireline* multicast problems [4, 5, 58]. A multicast problem comprises one or more source nodes (at which information is generated), several destinations (that demand all information available at the source nodes), relay nodes and directed communication channels between some nodes. Each channel is statistically independent of all other channels and the communication between different nodes is done through physically separated channels (wires). This means that the communication between two nodes does not affect the communication between other nodes. In this setup, the maximal achievable rate is

given by the minimal cut-capacity over all cuts separating the source nodes and a destination node. Because of the special structure of wireline networks, the cut-capacity for any cut is equal to the sum of the capacities of the channels crossing the cut.

This remarkable result for wireline networks is proved by performing separate channel and network coding in the network. First, we perform channel coding on each link of the network, so as to make it operate error-free at any rate below its capacity. This way, the problem is transformed into a flow problem in a graph where the capacity of each edge is equal to the information-theoretic capacity of the corresponding channel in the original network. If there is only one destination node, standard routing algorithms for finding the max-flow (min-cut) in graphs [66] achieve the capacity. However, when the number of destinations is more than one, these algorithms can fail. The key idea in [4] is to perform coding at the relay nodes. By [5, 58], linear codes are sufficient to achieve the capacity in multicast problems. These ideas are formulated in an algebraic framework and generalized to some other special network problems in [58]. Since then, there has been a lot of research on the benefits of coding over traditional routing schemes in networks from different viewpoints such as network management, security, etc. [6, 67]. In a wireless setup, however, the problem of finding the capacity region is more complicated. The main reason is that unlike wireline networks, in which communication between different nodes is done using separated media, in a wireless system the communication medium is shared. Hence, all transmissions across a wireless network are *broadcast*. Also any communication between two users can cause *interference* to the communication of other nodes. These two features, broadcast and interference, present new issues and challenges for performance analysis and system design. The capacity regions of many information-theoretic channels that capture these effects are not known. For instance, the capacity region for general broadcast channels is an unsolved problem [7]. The capacity of general relay channels is not known. However, there are some achievable results based on block Markov encoding and random binning [8]. These ideas have been generalized and applied to a multiple relay setup in [9, 10].

In this chapter we look at a special class of wireless networks that only incorporates

the broadcast feature of wireless networks.[1]We model each communication channel in the network as a memoryless erasure channel. We will often assume that the erasure channels are independent; however, we show that the results also hold when the various erasure channels are correlated. We require that each node send out the same signal on each outgoing link. However, for reception we use a multiple access model without interference, i.e., messages coming into a node from different incoming links do not interfere. In general, this is not true for a wireless system. However, this can be realized through some time, frequency or code division multiple access scheme. This simplification is important in making the solution of the problem tractable. Even the capacity of a single relay channel is not known.

Finally, we assume that complete side-information regarding erasure locations on each link is available to the destination (but not to the relay) nodes. If we assume that the erasure network operates on long packets, i.e., packets are either erased or received exactly on each link, then this assumption can be justified by using headers in the packets to convey erasure locations or by sending a number of extra packets containing this information. By making the packets very long, the overhead of transmitting the erasure locations can be made negligible compared to the packet length. We should remark that provided that the side-information is available to the destinations, all the results in this chapter hold for any packet length.

We should mention that our model is appropriate for wireless networks where all information transmission is packetized and where some form of interference-avoidance is already in place. Channel coding within each packet can be used to make each link behave as a packet erasure channel. Although our model does not incorporate interference (primarily because it is not clear what interference means for erasure channels) one way, perhaps, to account for interference is to allow the erasure channels coming into any particular node to be correlated (something that is permitted in our model).

The main result is that a max-flow min-cut type of result holds for multicast

[1][11, 12] have considered applications of network coding at the network layer for cost (energy) minimization in lossless wireless ad-hoc networks. In this chapter, we look at wireless features of the network in the physical layer.

problems in wireless erasure networks under the assumptions mentioned above. The definition of cut-capacity in these networks is such that it incorporates the broadcast nature of the network. We further show that similar to the wireline case, for multicast problems over wireless erasure networks, linear encoding at nodes achieves all the points in the capacity region. Working with linear encoding functions reduces the complexity of encoding and decoding. Building on the results presented here and using ideas from LT coding [13], it is shown in [14] that it is possible to reduce the delay incurred in the network. In their scheme, instead of using linear *block* codes, which is what we do here, the nodes send random linear combinations of their previously received signals at each time. This way nodes do not need to wait for receiving a full block before transmitting, which reduces the delay.

We once more need to emphasize the importance of the side-information on the erasure locations (or any other mechanism that provides the destination with the mapping from the source nodes to their incoming signals) for our result to hold. Interestingly, all the cut capacities of the network remain unchanged by making the above described side-information available to the receiver nodes. Thus, in some sense, what is shown in this chapter is that with appropriate side information made available to the receivers, the min-cut upper bound on capacity can be made tight. It would therefore be of further interest to see whether for other classes of networks it is possible to come up with the appropriate side-information to make the min-cut bounds tight.

This chapter is organized as follows. Section 4.2 defines notation used in this chapter and reviews some graph-theoretic definitions of importance. We introduce the network model in Section 4.3 and the problem setup in Section 4.4. Section 4.5 states the main result for multicast problems over wireless erasure networks with side-information available at destinations. Section 4.6 includes proofs of these results. Section 4.7 demonstrates the optimality of linear encoding. We mention future directions and conclude in Section 4.8.

$\mathcal{V}$	node set
$\mathcal{E}$	edge set
$\mathcal{S}$	the set of source nodes
$\mathcal{D}$	the set of destination nodes
$[\mathcal{V}_x, \mathcal{V}_y]$	$x-y$ cut-set described by x-set $\mathcal{V}_x$
X_i	symbol transmitted from node i
X_i^n	a transmitted block of n symbols from node i
Y_{ij}	channel output of edge (i,j)
Y_i	symbols received at node i from all incoming channels
$w^{(s)}$	message transmitted from source s
$\mathcal{W}^{(s)}$	message index set at source node s
$\hat{w}_{d_i}^{(s)}$	estimate at destination d_i of the message transmitted from s
$P_{d_i}^{(n)(s)}$	prob. of error in decoding source s at destination d_i

Table 4.1: Some important notation in this chapter

4.2 Preliminaries

4.2.1 Notation

Throughout this chapter, upper case letters (e.g., X, Y, Z) usually denote random variables and lower case letters (e.g., x, y, z) denote the values they take. Underlined letters (e.g., $\underline{x}$) are used to denote vectors. Sets are denoted by calligraphic alphabet (e.g., $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$). The complement of a set $\mathcal{A}$ is shown by $\mathcal{A}^c$. The transpose of matrix $\underline{x}$ is shown by $\underline{x}^\dagger$. $\exp(x)$ is used to denote 2^x.

Subscripts specify nodes, edges, inputs, outputs and time. For instance v_2 and X_3 could denote node number 2 and the output of node number 3 in the network respectively. Unless otherwise mentioned, commas are used to separate time subscripts from other subscripts. Superscripts are also used to refer to different sources. For example, $w^{(s)}$ could denote the message sent by node s.

Consider a sequence of numbers $x_1, x_2, x_3, \ldots$. We use notation x^n to denote the sequence $x_1, x_2, \ldots, x_n$. We also use notation $(x_i, \quad i \in \mathcal{I})$ to denote the ordered tuple specified by index set $\mathcal{I}$. Finally, $|\mathcal{X}|$ is the cardinality of set $\mathcal{X}$. Table 4.1 summarizes our notation.

4.2.2 Definitions for Directed Graphs

In this part, we briefly review the concepts and definitions from graph theory used in this chapter [73].

A directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, has vertex set $\mathcal{V}$ and directed edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. Without loss of generality, let

$$\mathcal{V} = \{1, 2, \ldots, |\mathcal{V}|\}.$$

We assume that the graph is finite, i.e., $|\mathcal{V}| < \infty$. For each node $v \in \mathcal{V}$, $\mathcal{N}_O(v)$ and $\mathcal{N}_I(v)$ are the set of edges leaving from and the set of edge going into v, respectively. Formally,

$$\begin{aligned} \mathcal{N}_O(v) &= \{(v,u)|(v,u) \in \mathcal{E}\} \\ \mathcal{N}_I(v) &= \{(u',v)|(u',v) \in \mathcal{E}\}. \end{aligned}$$

The *out-degree* $d_O(v)$ and *in-degree* $d_I(v)$ of v are defined as $d_O(v) = |\mathcal{N}_O(v)|$ and $d_I(v) = |\mathcal{N}_I(v)|$. A sequence of nodes $v_0, v_1, \ldots, v_n$ such that $(v_0, v_1), (v_1, v_2), \ldots, (v_n, v_0)$ are all in $\mathcal{E}$ is called a cycle. An acyclic graph is a directed graph with no cycles.

An $x - y$ *cut* for $x, y \in \mathcal{V}$ is a partition of $\mathcal{V}$ into two subsets $\mathcal{V}_x$ and $\mathcal{V}_y = \mathcal{V}_x^c$ such that $x \in \mathcal{V}_x$ and $y \in \mathcal{V}_y$. The *x-set* $\mathcal{V}_x$ (or *y-set* $\mathcal{V}_y$) determines the cut uniquely. For the $x - y$ cut given by $\mathcal{V}_x$, the *cut-set* $[\mathcal{V}_x, \mathcal{V}_y]$ is the set of edges going from the *x-set* to *y-set*, i.e.,

$$[\mathcal{V}_x, \mathcal{V}_y] = \{(u,v)|(u,v) \in \mathcal{E}, u \in \mathcal{V}_x, v \in \mathcal{V}_y\}.$$

We also define $\mathcal{V}_x^*$ as

$$\mathcal{V}_x^* = \{v|\exists\, u \text{ s.t. } (v,u) \in [\mathcal{V}_x, \mathcal{V}_y]\}.$$

$\mathcal{V}_x^*$ is the set of nodes in the x-set that has at least one of its outgoing edges in the cut-set.

Example 4.2.1. Consider the acyclic directed graph shown in Figure 4.1. $\mathcal{V} = \{1, 2, 3, 4\}$ is the set of nodes and $\mathcal{E} = \{(1,2), (3,2), (1,3), (3,4), (2,4)\}$ is the set of

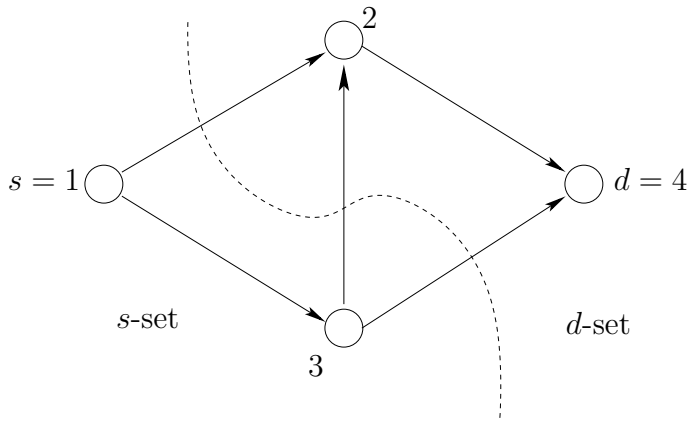


Figure 4.1: A directed acyclic graph with four nodes and five edges. The cut-set $\{(3,4),(3,2),(1,2)\}$ is shown by the dashed line.

edges. The source and destination nodes are $s = 1$ and $d = 4$, respectively. The *out-degree* of node 3 is 2, i.e., $d_O(3) = 2$. Looking at the $s - d$ cut specified by *s-set* $\mathcal{V}_s = \{1, 3\}$, the cut-set $[\mathcal{V}_s, \mathcal{V}_d]$ is the set $\{(3,4),(3,2),(1,2)\}$ and $\mathcal{V}_s^* = \{1, 3\}$.

4.3 Network Model

Wireless Packet Erasure Networks

We model the wireless packet[2] erasure network by a directed acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Each edge $(i, j) \in \mathcal{E}$ represents a memoryless packet erasure channel from node i to node j. For most of this chapter, we assume that erasure events across different links are independent. However, as described later in the chapter, the results go through for correlated erasure events. For independent erasure events, a packet sent across link (i, j) is either erased with probability of erasure ϵ_{ij} or received without error. We denote the input alphabet (the set of possible packets) of the erasure channel by $\mathcal{X}$.[3]

Let $Z_{ij,t}$ be a random variable indicating erasure occurrence across channel (i, j) at time t. For independent erasure events, $Z_{ij,t}$ has a Bernoulli distribution with

[2]Throughout this chapter a packet can be of any length. When the length of packets is one, the channel is a binary erasure channel.

[3]For simplicity and without loss of generality we consider $\mathcal{X} = \{0, 1\}$ in our analysis and proofs. However, we should remark that all the results and analysis hold for an input alphabet of arbitrary length.

parameter ϵ_{ij}. If an erasure occurs on link $(i,j) \in \mathcal{E}$ at time t, the value of $Z_{ij,t}$ will be one, otherwise $Z_{ij,t}$ will be zero. Note that the behavior of the network can be fully determined by the values of $Z_{ij,t}$ for all links and all times and the operation performed at each node.

We assume that transmissions on each channel experience one unit of time delay. The input of all the channels originating from node i is denoted by X_i chosen from input alphabet $\mathcal{X}$. Note that with this definition we have required that each node transmit the same symbol on all its outgoing edges, i.e., all channels corresponding to edges in $\mathcal{N}_O(i)$ have the (same) input X_i (See Figure 4.2.) This constraint incorporates broadcast in our network model. The output of the communication channel corresponding to edge $(i,j) \in \mathcal{E}$ is denoted by Y_{ij}; Y_{ij} lies in output alphabet $\mathcal{Y} = \mathcal{X} \cup \{e\}$, where e denotes the erasure symbol. We also assume that the outputs of all channels corresponding to edges in $\mathcal{N}_I(i)$ are available at node i. This condition is equivalent to having no interference in receptions in the network. Having this, let $Y_i = (Y_{ji},\ (j,i) \in \mathcal{N}_I(i))$ be the symbols that are received at node i from all its incoming channels. We have $Y_i \in \prod_{j:(j,i)\in\mathcal{E}} \mathcal{Y}$. The relation between the Y_is and X_is defines a coding scheme for the network.

Based on the properties of the network mentioned above, if we consider the inputs and outputs up to time t, then the conditional probability function of the outputs of all the channels (edges) up to time t given all the inputs of all the channels up to time t and all the previous outputs, can be written as follows for all t:

$$\begin{aligned}&\mathrm{P}\left((y_{ij,t},\ (i,j)\in\mathcal{E})\middle|(x_l^t,\ l\in\mathcal{V}),(y_{ij}^{t-1},\ (i,j)\in\mathcal{E})\right)\\&=\mathrm{P}\left((Y_{ij}=y_{ij,t},\ (i,j)\in\mathcal{E})|(X_l=x_{l,t},\ l\in\mathcal{V})\right).\end{aligned}$$

For independent erasure events, we further have

$$\mathrm{P}\left((y_{ij,t},\ (i,j)\in\mathcal{E})\middle|(x_l^t,\ l\in\mathcal{V}),(y_{ij}^{t-1},\ (i,j)\in\mathcal{E})\right)=\prod_{i\in\mathcal{V}}\prod_{j:\,(i,j)\in\mathcal{N}_O(i)}\mathrm{P}(Y_{ij}=y_{ij,t}|X_i=x_{i,t}). \tag{4.1}$$

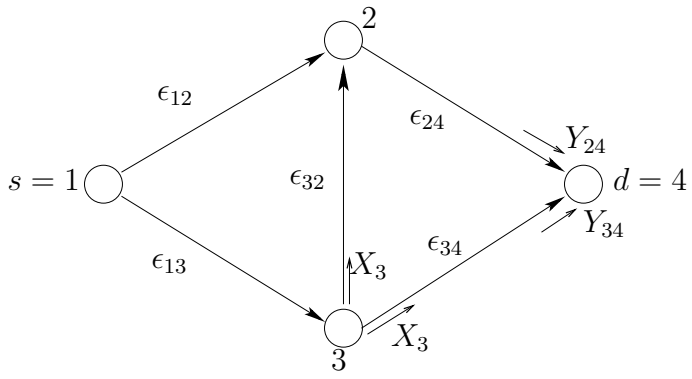


Figure 4.2: (i) An erasure wireless network with the graph representation of example 4.2.1. Probability of erasure on link (i, j) is ϵ_{ij}. Each node (e.g., node 3) transmits the same signal (X_3) across its outgoing channels. Since the network is interference-free, node 4 receives both signals Y_{24} and Y_{34} completely. (ii) In this network, cut-capacity for s-set $\mathcal{V}_s = \{1, 3\}$ is $C(\mathcal{V}_s) = 1 - \epsilon_{12} + 1 - \epsilon_{32}\epsilon_{34}$.

Multicast Problem

In this chapter, we consider a class of network problems called multicast problems. Any network problem is characterized by a collection of information sources, a collection of source nodes at which one or more information sources are available, and a collection of destination nodes. Each destination node demands a subset of information sources. The class of network problems that we consider in this chapter is the multiple source/multiple destination multicast, where each of the destinations demands all of the information sources. This problem can be further specified by the following sets:

- $\mathcal{S} = \{s_1, s_2, \ldots, s_{|\mathcal{S}|}\} \subset \mathcal{V}$ denotes the information source nodes. We assume that each of the source nodes generates an information (message) which is modeled by an i.i.d. uniformly distributed random process. Information sources at different nodes are assumed to be independent.

- $\mathcal{D} = \{d_1, d_2, \ldots, d_{|D|}\} \subset \mathcal{V}$ denotes the set of destination nodes.

Note that $\mathcal{S} \cap \mathcal{D}$ may not be empty, i.e., a node can be a destination node for one information source and a source node for another. Also destination nodes can act as relay nodes for other destination nodes in the network.

Side-information at Destinations

In most parts of the chapter we assume that each destination node $d \in \mathcal{D}$ has complete knowledge of the erasure locations on each link of the network that is on a path from the source set to d. In other words, d knows values of the $z_{ij,t}$, for all $(i,j) \in \mathcal{E}$ and all times t, for which (i,j) is on at least one path from one of the sources to d. This serves as channel side-information provided to the destinations from across the network. In the case when we consider large packets (alphabet), this side-information can be provided using negligible overhead. More discussion of this model appears in Section 8.

Cut-capacity Definition

Consider an $s-d$ cut given by *s-set* $\mathcal{V}_s$ as defined in Section 4.2.2. We define $X(\mathcal{V}_s)$ and $Y(\mathcal{V}_s)$ as

$$\begin{aligned} X(\mathcal{V}_s) &= \{X_i | i \in \mathcal{V}_s^*\} \\ Y(\mathcal{V}_s) &= \{Y_{ij} | (i,j) \in [\mathcal{V}_s, \mathcal{V}_s^c]\}. \end{aligned} \tag{4.2}$$

At the end of this section, we define the cut-capacity for wireless erasure networks. In wireline networks, the value of the cut-capacity is the sum of the capacities of the edges in the cut-set [58]. Such a definition of cut-capacity in wireline networks makes sense because the nodes can send out different signals across their outgoing edges. However, this is not the case for wireless erasure networks where broadcast transmissions are required. The following definition of cut-capacity is different from that in the wireline network settings, and it incorporates the broadcast nature of transmission in our network.

Definition 1. *Consider an erasure wireless network represented by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and probabilities of erasure ϵ_{ij} as described in Section 4.3. Let s and d_l be the source and destination nodes, respectively. The cut-capacity corresponding to any $s - d_l$ cut*

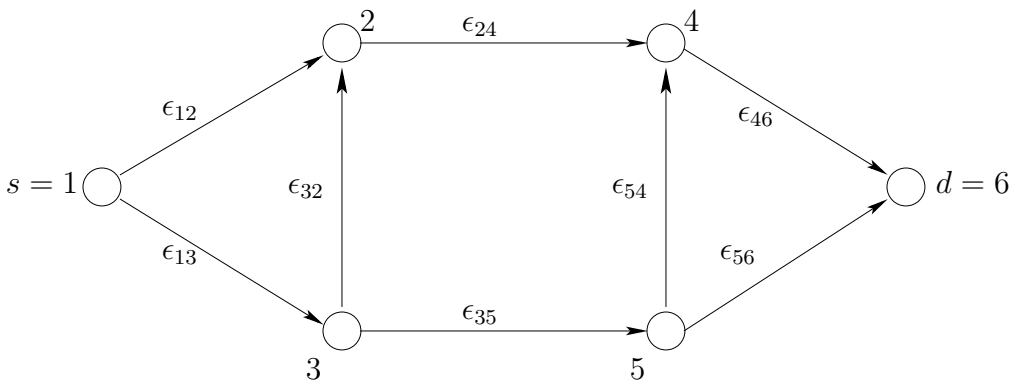


Figure 4.3: For the cut-set specified by the s-set $\mathcal{V}_s = \{1, 3, 4\}$ the cut-capacity is $C(\mathcal{V}_s) = 1 - \epsilon_{12} + 1 - \epsilon_{46} + 1 - \epsilon_{35}\epsilon_{32}$.

represented by s-set, $\mathcal{V}_s$ is denoted by $C(\mathcal{V}_s)$ and is equal to

$$C(\mathcal{V}_s) = \sum_{i \in \mathcal{V}_s^*} \left(1 - \prod_{j:\,(i,j) \in [\mathcal{V}_s, \mathcal{V}_{d_l}]} \epsilon_{ij} \right). \tag{4.3}$$

Example 4.3.1. Consider the network represented by the directed graph of example 4.2.1. (See Figure 4.2.) For the $s - d$ cut specified by the s-set $\mathcal{V}_s = \{1, 3\}$, the cut-capacity is

$$C(\mathcal{V}_s) = 1 - \epsilon_{12} + 1 - \epsilon_{32}\epsilon_{34}.$$

Looking at this example, we see that all edges in the cut-set that originate from a common node, i.e., edges $(3, 2)$ and $(3, 4)$, together contribute a value of one minus the product of their erasure probabilities, i.e., $1 - \epsilon_{32}\epsilon_{34}$ to the cut-capacity. This observation holds in general for wireless erasure networks.

Example 4.3.2. As another example, consider the network shown in Figure 4.3 with one source $s = 1$ and one destination $d = 6$. The cut-capacity corresponding to the $s - d$ cut specified by $\mathcal{V}_s = \{1, 3, 4\}$ is $C(\mathcal{V}_s) = 1 - \epsilon_{12} + 1 - \epsilon_{46} + 1 - \epsilon_{35}\epsilon_{32}$.

4.4 Problem Statement

We next define the class of block codes considered in this chapter. A $(\lceil 2^{nR_1} \rceil, \ldots, \lceil 2^{nR_{|\mathcal{S}|}} \rceil, n)$ code for the multicast problem in a wireless erasure network described in the previous sections, consists of the following components:

- A set of integers $\mathcal{W}^{(s_i)} = \{1, 2, \ldots, \lceil 2^{nR_i} \rceil\}$ for each source node $s_i \in \mathcal{S}$. $\mathcal{W}^{(s_i)}$ represents the set of message indices corresponding to node s_i. $w^{(s)}$ denotes the message of source $s \in \mathcal{S}$. We assume that the messages are equally likely and independent.

- A set of encoding functions $\{f_{i,t}\}_{t=1}^{n}$ for each node $i \in \mathcal{V}$, where

$$x_{i,t} = f_{i,t}(w^{(i)}, y_i^{t-1})$$

  is the signal transmitted by node i at time t. Note that $x_{i,t}$ is a function of the message $w^{(i)}$ that node $i \in \mathcal{V}$ wants to transmit in the current block [4] and all symbols received so far by node i from its incoming channels. If i is not a source node, we set $w^{(i)} = 0$ for all blocks and all times.

- A decoding function g_{d_i} at destination node $d_i \in \mathcal{D}$,

$$g_{d_i} : \mathcal{W}^{(d_i)} \times \mathcal{Y}_{d_i}^{n} \times \{0,1\}^{n|\mathcal{E}|} \rightarrow \prod_{s \in \mathcal{S}} \mathcal{W}^{(s)}$$

  such that

$$\underline{\hat{w}}_{d_i} = (\hat{w}_{d_i}^{(s)}, \quad s \in \mathcal{S}) = g_{d_i}(w^{(d_i)}, y_{d_i}^{n}, (z_{ij,t}\,, (i,j) \in \mathcal{E},\, 1 \leq t \leq n)) \tag{4.4}$$

  where $\hat{w}_{d_i}^{(s)}$ is the estimate of the message sent from source $s \in \mathcal{S}$ based on received signals at d_i, information source available at d_i[5], $w^{(d_i)}$, and also the erasure occurrences on all the links of the network in the current block.

[4] The value of $w^{(i)}$ does not change in one block.

[5] If $d_i \notin \mathcal{S}$ without loss of generality we set $w^{(d_i)} = 0$ and $\mathcal{W}^{(d_i)} = \{0\}$ for all blocks.

Note that X_i, Y_{ij} and Y_i all depend on the message vector $\underline{w} = (w^{(s)},\, s \in \mathcal{S})$, that is being transmitted. Therefore we will write them as $X_i(\underline{w})$,$Y_{ij}(\underline{w})$ and $Y_i(\underline{w})$ to specify what specific set of messages is transmitted.

Associated with every destination node $d \in \mathcal{D}$ and every information source $s \in \mathcal{S}$ is a probability that the message will not be decoded correctly.[6]

$$P_d^{(n)(s)} = \Pr(\hat{W}_d^{(s)} \neq W^{(s)}), \tag{4.5}$$

where $P_d^{(n)(s)}$ is defined under the assumption that all the messages are independent and uniformly distributed over $\mathcal{W}^{(s)}$, $s \in \mathcal{S}$. The set of rates $(R_s,\, s \in \mathcal{S})$ is said to be achievable if there exists a sequence of $(\lceil 2^{nR_1} \rceil, \ldots, \lceil 2^{nR_{|\mathcal{S}|}} \rceil, n)$ codes such that $P_d^{(n)(s)} \to 0$ as $n \to \infty$ for all $s \in \mathcal{S}$ and $d \in \mathcal{D}$. The capacity region is the closure of the set of achievable rates.

4.5 Main Results

In this section we present the main results of this chapter

Theorem 4.1. *Consider a single source/single destination wireless erasure network described by the directed acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and the assumptions of Section 4.3. Let $s \in \mathcal{V}$ and $d \in \mathcal{V}$ denote the network's source and destination, respectively. Then the capacity of the network with side-information at the destination is given by the value of the minimum value s-d cut. More precisely, we have*

$$C = \min_{\mathcal{V}_s : \mathcal{V}_s \text{an } s-d \text{ cut}} C(\mathcal{V}_s). \tag{4.6}$$

Remark 1. *The results derived in this chapter are stated for erasure wireless networks with broadcast property (and no interference). However, based on the results of this chapter, it is possible to derive the capacity of multicast problems over* error-free *networks (with the broadcast property and without interference), with or without*

[6]Note that if d is a source node, we assume without loss of generality that $P_d^{(n)(d)} = 0$.

capacitated links.

Remark 2. *Although we have assumed that the erasure events across the network are independent, the capacity results of this chapter also hold for the case when the erasure events are correlated, i.e.* Z_{ij}, $(i,j) \in \mathcal{E}$ *are dependent on each other. In that case the definition of the cut capacity should be modified as described in (4.22). (See Remark 4 in Appendix A.)*

Example 4.5.1. Recall the single source/single destination network of example 4.3.1. (See Figure 4.2.) By Theorem 4.1, the capacity of this network is

$$C = \min\{1-\epsilon_{12}+1-\epsilon_{32}\epsilon_{34}, 1-\epsilon_{34}+1-\epsilon_{24}, 1-\epsilon_{12}\epsilon_{13}, 1-\epsilon_{13}+1-\epsilon_{24}\}$$

The following theorems generalize the single source/single destination result to general multicast problems.

Theorem 4.2. *Consider a multiple source/single destination wireless erasure network described by directed acyclic graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ *and the assumptions of Section 4.3. Suppose that the destination requests all of the information from all of the sources. Let* $\mathcal{S} \subset \mathcal{V}$ *and* $d \in \mathcal{V}$ *denote the set of source nodes and the destination node, respectively. The capacity region of the network with side-information provided at the destination is given by*

$$C(\mathcal{G}, \mathcal{S}, d) \stackrel{\triangle}{=} \left\{ (R_s,\, s \in \mathcal{S}) \middle| 0 \leq \sum_{s \in \mathcal{V}' \cap \mathcal{S}} R_s \leq C(\mathcal{V}') \qquad \forall\ \mathcal{V}' \subset \mathcal{V} - \{d\} \right\}. \tag{4.7}$$

In other words, the total rate of information transmission to d *across any cut* $[\mathcal{V}', \mathcal{V}_d]$*, should not exceed the cut-capacity of that cut.*

Example 4.5.2. Consider the network shown in Figure 4.4 with two sources $\{1,2\}$ and one destination $\{3\}$. Then according to Theorem 4.2, the capacity region is

$$\{(R_1, R_2) \in \mathbb{R}^+ \times \mathbb{R}^+ | R_1 \leq 1-\epsilon_{12}\epsilon_{13}, R_2 \leq 1-\epsilon_{23}, R_1+R_2 \leq 1-\epsilon_{23}+1-\epsilon_{13}\}.$$

Theorem 4.3. *Consider a multicast problem with multiple sources and multiple destinations. Let $\mathcal{S}, \mathcal{D} \subset \mathcal{V}$ denote the set of source nodes, and destination nodes, respectively. The capacity region of the network with side-information is given by the intersection of the capacity regions of the multicast problem between the sources and each of the destinations, i.e,*

$$C(\mathcal{G}, \mathcal{S}, \mathcal{D}) = \bigcap_{d \in \mathcal{D}} C(\mathcal{G}, \mathcal{S}, d). \tag{4.8}$$

Corollary 4.4. *Consider a multicast problem with one source denoted by s and multiple destinations denoted by $d_1, \ldots, d_{|\mathcal{D}|}$. The capacity of the network is given by the minimum value of the cuts between the source node and any of the destinations, i.e.,*

$$C = \min_{d_i \in \mathcal{D}} \min_{\mathcal{V}_s : s - d_i \, cut} C(\mathcal{V}_s).$$

Example 4.5.3. Consider the network shown in Figure 4.2. Suppose that we are decoding at node 2 and 4, i.e., $\mathcal{D} = \{2, 4\}$. Based on Corollary 4.4, the capacity of this network is

$$C = \min\{1 - \epsilon_{12} + 1 - \epsilon_{32}, 1 - \epsilon_{34} + 1 - \epsilon_{24}, 1 - \epsilon_{12}\epsilon_{13}, 1 - \epsilon_{13} + 1 - \epsilon_{24}\}.$$

The above results show that the capacity region for multicast problems over wireless erasure networks has a max-flow min-cut interpretation. This result is similar to multicast problems in wireline networks [4], however the definition of the cut-capacity is different. Recall from [4] that in wireline networks, the cut-capacity is the sum of the capacities of the links in the cut-set. Since wireless erasure networks incorporate broadcast, the cut-capacity is the sum of the capacities of each broadcast system that operates across the cut.

The next theorem states that linear network coding is sufficient for achieving the capacity region.

Theorem 4.5. *Consider a multicast problem with multiple sources and multiple desti-*

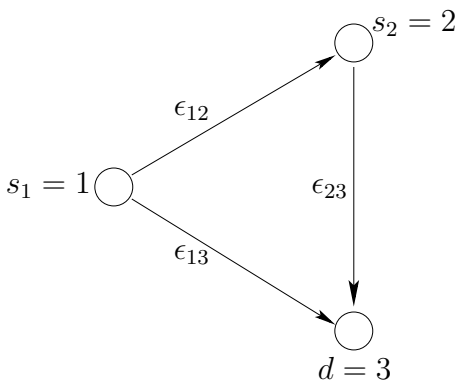


Figure 4.4: A wireless erasure network with two source, $\mathcal{S} = \{1, 2\}$ and one destination, $\mathcal{D} = \{3\}$.

nations. Then any rate vector in the capacity region $C(\mathcal{G}, \mathcal{S}, \mathcal{D})$ of the network defined in Theorem 4.3 is achievable with linear block coding.

In the next section, we prove Theorems 4.1, 4.2 and 4.3. In Section 4.7, we look at the performance of the network using random linear coding and prove Theorem 4.5.

4.6 Proof of Theorems

4.6.1 Proof of Theorems 4.1 and 4.2

In this section we prove the results stated for multi-source/ single destination network problems. We start by proving the converse.

4.6.1.1 Converse

We prove the converse part by considering perfect cooperation among subsets of nodes. Consider the cut specified by d-set $\mathcal{V}_d$. Let all of the nodes in $\mathcal{V}_d$ and all of the nodes in $\mathcal{V}_d^c$ cooperate perfectly, i.e., each node has access to all of the information known to nodes in its set. In this case, we have a multiple input, multiple output point-to-point erasure channel. Consider all source nodes in $\mathcal{V}_d^c$. Then clearly, the sum-rate of these source nodes must be less than the capacity of the multiple input

multiple output point-to-point erasure channel. The capacity of this point-to-point communication channel is

$$C_{col} = \max_{P(x_i,\quad i\in\mathcal{V}_d^{c*})} I((X_i,\, i \in \mathcal{V}_d^{c*}); (Y_{ij},\, (i,j) \in [\mathcal{V}_d^c, \mathcal{V}_d])).$$

Since the channels are independent and memoryless, the mutual information is maximized when the different X_is are i.i.d. and uniform on the input alphabet $\mathcal{X}$. In this case, the above mutual information equals the cut-capacity corresponding to the cut-set $[\mathcal{V}_d^c, \mathcal{V}_d]$, i.e.,

$$C_{col} = C(\mathcal{V}_d^c).$$

Therefore, for any cut-set $[\mathcal{V}_d^c, \mathcal{V}_d]$ the sum-rate of the information sources in set $\mathcal{V}_d^c$ satisfies

$$\sum_{s\in\mathcal{S}\cap\mathcal{V}_d^c} R_s \leq C_{col} = C(\mathcal{V}_d^c).$$

The complete analysis appears in Appendix 4.9.1. The proof follows the same lines as the max-flow min-cut upper bound of Cover and Thomas for multi-terminal networks [71, Sec. 14.10].

4.6.1.2 Achievability

In this section we prove that all of the rates arbitrarily close to rates in the capacity regions given in Theorems 4.1 and 4.2 are achievable for a multiple sources/single destination multicast problem. We next use random coding techniques to show this result.

We employ random block codes in the network. Each node transmits the next block of n symbols only after it has received all n symbols corresponding to the present block from each of its incoming channels. Let $L_{\max}$ denote the length of the longest path from a source to the destination in the network. Since each transmission introduces one unit of time delay, the maximal delay between the transmission of a message from one source and to its receipt at the destination using block codes of length n is $nL_{\max}$. We do not use any information from previously decoded blocks

to decode the current set of messages. Also note that since our model assumes that the reception is interference-free, there is no confusion up among different blocks at any node. Therefore, if the network operates for nB units of time (i.e., B blocks of length-n symbols) then the destination has received all of the information required for decoding the $B - L_{\max}$ first messages transmitted from each source $s \in \mathcal{S}$, i.e., $w_b^{(s)}, \quad b = 1, \ldots, B - L_{\max}$. Since the network size is finite, as $B \to \infty$, for fixed n, the rate $R_s \frac{B-L_{\max}}{B}$ approaches R_s.[7]

The same codebook and encoding and decoding functions are used for all the blocks. We explain the coding scheme for transmitting one set of messages from the sources to the destination. Below we describe the encoding and decoding processes.

- **Codebook Generation and Encoding:** For each node $i \in \mathcal{V}$, the encoding function

$$f_i : \mathcal{W}^{(i)} \times \mathcal{Y}_i^n \to \mathcal{X}^n$$

is generated randomly as follows. For each $y^n \in \mathcal{Y}_i^n$ and for each $w^{(i)} \in \mathcal{W}^{(i)}$ we draw the symbols of $f_i(w^{(i)}, y^n) \in \mathcal{X}^n$ randomly and independently according to a binary Bernoulli distribution with parameter $1/2$. Thus the channel input at node $i \in \mathcal{V}$ is $X_i^n = f_i(w^{(i)}, y^n)$ when the message at node i is $w^{(i)}$ and the incoming sequence is $y^n \in \mathcal{Y}_i^n$. The destination has perfect knowledge of all the encoding functions $f_i(.), \quad i \in \mathcal{V}$ thus generated.[8]

- **Decoding:** The destination "simulates" the network to decode the messages. Suppose that message vector $\underline{w}_0 = (w_0^{(s)}, \quad s \in \mathcal{S})$ is transmitted and $y_d^n(\underline{w}_0)$ is received at destination d. By assumption, the receiver is knows the erasure locations on all the links of the network, i.e., $(z_{ij}^n, \ (i,j) \in \mathcal{E})$. Having all of the

[7]We could also consider the case when different sources transmit different numbers of messages in B block uses. In that case, if L_s denotes the longest path from $s \in \mathcal{S}$ to the destination, we could transmit $B - L_s$ messages from information source s to the destination. However, for simplicity of notation and analysis we assume that all of the nodes send the same number of messages in a synchronized fashion.

[8]Note that the encoding functions thus constructed satisfy a causality condition that is more strict than what is defined in Section 4.4. Here each transmitted block is only a function of the immediately previous block of received symbols. In Section 4.3, each transmitted symbol could be a function of all previous symbols.

erasure locations and all of the encoding functions applied at different nodes in the network,[9] the destination can compute the values of $X_i^n(\underline{w}), Y_{ij}^n(\underline{w})$ and $Y_i^n(\underline{w})$ for all nodes and edges for any $\underline{w} \in \prod_{s\in\mathcal{S}} \mathcal{W}^{(s)}$. If there exists a unique message vector $\underline{w} \in \prod_{s\in\mathcal{S}} \mathcal{W}^{(s)}$ such that the computed value of $Y_d^n(\underline{w})$ equals the value $y_d^n(\underline{w}_0)$ of the received signal at the destination, then $\underline{w}$ is declared as the decoder output. Otherwise, the decoder declares an error.

Since the computed value of $Y_d^n(\underline{w}_0)$ for transmitted message $\underline{w}_0$ always matches the received signal at the destination, an error occurs if and only if there is another message vector $\underline{w} \neq \underline{w}_0$ for which $Y_d^n(\underline{w}) = Y_d^n(\underline{w}_0) = y_d^n(\underline{w}_0)$. In the next section we compute the probability of this event and show that for large blocks this probability can be made arbitrarily close to zero provided that the rate vector $(R_s,\ s \in \mathcal{S})$ is inside the capacity region described in Theorems 4.1 and 4.2.

4.6.1.3 Probability of Error

Let $\Pr(err)$ be the probability of error averaged over all possible functions f_i. In other words, if $P_e^{(n)}$ is the probability that $\hat{w}_0^{(s)}$, the destination's estimate of the transmitted message $\underline{w}_0$, is not equal to $\underline{w}_0$, then $\Pr(err)$ is the expected value of $P_e^{(n)}$ over all possible encoding functions at all nodes.[10] More precisely,

$$P_e^{(n)} = \Pr(\exists s \in \mathcal{S} \text{ s.t. } \hat{w}_0^{(s)} \neq w_0^{(s)}),$$

and $\Pr(err) = \mathrm{E}\, P_e^{(n)}$. Because of the symmetry of the code construction

$$\Pr(err) = \Pr(err | \underline{W} = \underline{w}_0 \text{ is transmitted}) \tag{4.9}$$

where $\underline{W} = (W^{(s)}, \quad s \in \mathcal{S})$. Therefore we will find the average probability of error when message vector $\underline{w}_0$ is transmitted from the sources. Recall the notation $X_i^n(\underline{w}_0)$ and $Y_i^n(\underline{w}_0)$ and $Y_{ij}^n(\underline{w}_0)$, $(i,j) \in \mathcal{E}$. For each $\underline{w} \in \underline{\mathcal{W}} \stackrel{\triangle}{=} \prod_{s\in\mathcal{S}} \mathcal{W}^{(s)}$, $\underline{w} \neq \underline{w}_0$, define

[9] We also assume that the destination knows the topology of the network.

[10] Note that if $P_e^{(n)}$ goes to zero as n grows larger, so will $P_d^{(n)(s)}$ of (4.5) for every $s \in \mathcal{S}$.

the following event:

$$E(\underline{w}) = \{Y_d^n(\underline{w}) = Y_d^n(\underline{w}_0)\}. \tag{4.10}$$

Let $\mathcal{A}_\delta^{(n)}(i)$ be the event that the erasure locations on the channels going out of node i are jointly δ-strongly typical, i.e.,

$$\mathcal{A}_\delta^{(n)}(i) = \{(z_{ij}^n,\, j : (i,j) \in \mathcal{E}) \quad \text{are jointly } \delta\text{-strongly typical}\}$$

[1, eq. (13.107)] and define

$$\mathcal{A}_\delta^{(n)} = \bigcap_{i=1}^{|\mathcal{V}|} \mathcal{A}_\delta^{(n)}(i).$$

Note that by the weak law of large numbers [71], $\Pr(\mathcal{A}_\delta^{(n)}(i)) \to 1$ as $n \to \infty$, and hence for all $\delta > 0$

$$\Pr(\mathcal{A}_\delta^{(n)}) \geq 1 - |\mathcal{V}|\delta, \quad \text{for } n \text{ sufficiently large.}$$

Using the definition of the above events, $\Pr(err)$ can be written as

$$\begin{aligned}
\Pr(err) &= \Pr(err|\underline{W} = \underline{w}_0) \\
&= \Pr\left(\bigcup_{\underline{w} \in \underline{\mathcal{W}} - \{\underline{w}_0\}} E(\underline{w})\right) \\
&= \Pr\left(\bigcup_{\underline{w} \in \underline{\mathcal{W}} - \{\underline{w}_0\}} E(\underline{w}) | \mathcal{A}_\delta^{(n)}\right) \Pr(\mathcal{A}_\delta^{(n)}) + \Pr\left(\bigcup_{\underline{w} \in \underline{\mathcal{W}} - \{\underline{w}_0\}} E(\underline{w}) | \mathcal{A}_\delta^{(n)^c}\right) \Pr(\mathcal{A}_\delta^{(n)^c}) \\
&\leq \sum_{\underline{w} \in \underline{\mathcal{W}} - \{\underline{w}_0\}} \Pr(E(\underline{w}) | \mathcal{A}_\delta^{(n)}) + |\mathcal{V}|\delta. \qquad (4.11)
\end{aligned}$$

Therefore, using strong typicality and the union bound on the probability of events, we only look at network instantiations that are "strongly typical." We next bound the conditional probability of $E(\underline{w})$ given $\mathcal{A}_\delta^{(n)}$.

Corresponding to each cut in the network, represented by *d-set* $\mathcal{V}_d \ni d$, define the

following event:

$$B[\mathcal{V}_d] = \Big(\bigcap_{i \in \mathcal{V}_d} \{Y_i^n(\underline{w}) = Y_i^n(\underline{w}_0)\} \Big) \bigcap \Big(\bigcap_{i \in \mathcal{V}_d^c} \{Y_i^n(\underline{w}) \neq Y_i^n(\underline{w}_0)\} \Big) \tag{4.12}$$

The interpretation of the above event is as follows. By definition of $E(\underline{w})$, we know that the received signal at the destination is the same for $\underline{w}$ and $\underline{w}_0$, but $\underline{w} \neq \underline{w}_0$. Therefore we can partition the nodes of the network into two sets: the "distinguishable" and the "indistinguishable" set. The "distinguishable" set contains all nodes for which the signal received at those nodes when $\underline{w}$ is transmitted differs from the signal received when $\underline{w}_0$ is transmitted. All the other nodes, for which the received signals for $\underline{w}$ and $\underline{w}_0$ are the same, are in the "indistinguishable" set. Clearly, these two sets define a cut. Event $B[\mathcal{V}_d]$ corresponds to the case when the "indistinguishable" set (containing d) is equal to $\mathcal{V}_d \subset \mathcal{V}$. Note that these events are all disjoint and also $E(\underline{w}) = \bigcup_{\mathcal{V}_d:\text{d-set}} B[\mathcal{V}_d]$.

Define

$$\mathcal{M}(\underline{w}) = \{s | s \in \mathcal{S}, w_0^{(s)} \neq w^{(s)}\} \tag{4.13}$$

to be the subset of source nodes for which the corresponding messages in $\underline{w}$ and $\underline{w}_0$ are different. Set $\mathcal{M}(\underline{w})$ is not empty since $\underline{w}_0 \neq \underline{w}$ by assumption. In what follows we bound the probability of event $B[\mathcal{V}_d]$ by considering the edges in the cut-set $[\mathcal{V}_x, \mathcal{V}_x^c]$ where $\mathcal{V}_x \stackrel{\triangle}{=} \mathcal{V}_d^c \cup \mathcal{M}(\underline{w})$. Note that $\mathcal{V}_x^c$ is a d-set since if the destination is a source of information, it is aware of the message it has transmitted and so $d \notin \mathcal{M}(\underline{w})$.

Consider any edge $(i, j) \in [\mathcal{V}_x, \mathcal{V}_x^c]$. We know that the transmitted signal $X_i^n = f_i(W^{(i)}, Y_i^n)$ from node i is a function of the message it wants to transmit , $W^{(i)}$, and the received signal at its incoming edges, Y_i^n. For any node i in $\mathcal{V}_d^c \cup \mathcal{M}(\underline{w})$, either the received signal Y_i^n or message $w^{(i)}$ is different for message vectors $\underline{w}$ and $\underline{w}_0$. Thus, for a randomly designed code, the transmitted signal by node i for message vector $\underline{w}$ is independent of the corresponding X_i^n for message vector $\underline{w}_0$. Using this observation

we next bound the probability of the event $E(\underline{w})$ conditioned on $\mathcal{A}_\delta^{(n)}$.

$$
\begin{aligned}
& \Pr(E(\underline{w})|\mathcal{A}_\delta^{(n)}) \\
= \; & \Pr\left(\bigcup_{\mathcal{V}_d:d\text{-}set} B[\mathcal{V}_d]|\mathcal{A}_\delta^{(n)}\right) = \sum_{\mathcal{V}_d:d\text{-}set} \Pr(B[\mathcal{V}_d]|\mathcal{A}_\delta^{(n)}) \\
= \; & \sum_{\mathcal{V}_d:d\text{-}set} \Pr\left(\left(\bigcap_{j\in\mathcal{V}_d}\{Y_j^n(\underline{w}) = Y_j^n(\underline{w}_0)\}\right)\bigcap\left(\bigcap_{i\in\mathcal{V}_d^c}\{Y_i^n(\underline{w}) \neq Y_i^n(\underline{w}_0)\}\right)|\mathcal{A}_\delta^{(n)}\right) \\
\overset{(a)}{\leq} \; & \sum_{\mathcal{V}_x:\mathcal{M}(\underline{w})\subset\mathcal{V}_x} \Pr\Big(\bigcap_{i,j:(i,j)\in[\mathcal{V}_x,\mathcal{V}_x^c]}\{(w^{(i)}, Y_i^n(\underline{w})) \neq (w_0^{(i)}, Y_i^n(\underline{w}_0)),\, Y_{ij}^n(\underline{w}) = Y_{ij}^n(\underline{w}_0)\}|\mathcal{A}_\delta^{(n)}\Big) \\
\overset{(b)}{=} \; & \sum_{\mathcal{V}_x:\mathcal{M}(\underline{w})\subset\mathcal{V}_x} \Pr\Big(\bigcap_{i\in\mathcal{V}_x^*}\bigcap_{j:(i,j)\in[\mathcal{V}_x,\mathcal{V}_x^c]}\{(w^{(i)}, Y_i^n(\underline{w})) \neq (w_0^{(i)}, Y_i^n(\underline{w}_0)),\, Y_{ij}^n(\underline{w}) = Y_{ij}^n(\underline{w}_0)\}|\mathcal{A}_\delta^{(n)}\Big) \\
\overset{(c)}{=} \; & \sum_{\mathcal{V}_x:\mathcal{M}(\underline{w})\subset\mathcal{V}_x} \Pr\left(\bigcap_{i\in\mathcal{V}_x^*}\bigcap_{j:(i,j)\in[\mathcal{V}_x,\mathcal{V}_x^c]}\{Y_{ij}^n(\underline{w}) = Y_{ij}^n(\underline{w}_0)\}\middle|\mathcal{A}_\delta^{(n)},\, \bigcap_{i\in\mathcal{V}_x^*}\{(w^{(i)}, Y_i^n(\underline{w})) \neq (w_0^{(i)}, Y_i^n(\underline{w}_0))\}\right) \\
& \qquad \cdot \Pr((w^{(i)}, Y_i^n(\underline{w})) \neq (w_0^{(i)}, Y_i^n(\underline{w}_0)),\, \forall\, i \in \mathcal{V}_s^*) \\
\leq \; & \sum_{\mathcal{V}_x:\mathcal{M}(\underline{w})\subset\mathcal{V}_x} \Pr\left(\bigcap_{i\in\mathcal{V}_x^*}\bigcap_{j:(i,j)\in[\mathcal{V}_x,\mathcal{V}_x^c]}\{Y_{ij}^n(\underline{w}) = Y_{ij}^n(\underline{w}_0)\}\middle|\mathcal{A}_\delta^{(n)},\, \bigcap_{i\in\mathcal{V}_x^*}\{(w^{(i)}, Y_i^n(\underline{w})) \neq (w_0^{(i)}, Y_i^n(\underline{w}_0))\}\right) \\
\overset{(d)}{=} \; & \sum_{\substack{\mathcal{V}_x:\mathcal{M}(\underline{w})\subset\mathcal{V}_x \\ d\notin\mathcal{V}_x}} \prod_{i\in\mathcal{V}_x^*} \Pr\left(\bigcap_{j:(i,j)\in[\mathcal{V}_x,\mathcal{V}_x^c]}\{Y_{ij}^n(\underline{w}) = Y_{ij}^n(\underline{w}_0)\}\middle|\mathcal{A}_\delta^{(n)},\, \{(w^{(i)}, Y_i^n(\underline{w})) \neq (w_0^{(i)}, Y_i^n(\underline{w}_0))\}\right).
\end{aligned}
\tag{4.14}
$$

Here

(a) follows since $\Pr(A, B) \leq \Pr(A)$ for any events A and B. Instead of looking at equalities on every edge and every node of the network, we are looking at the nodes having an edge from $[\mathcal{V}_x, \mathcal{V}_x^c]$ connected to them, where $\mathcal{V}_x = \mathcal{V}_d^c \cup \mathcal{M}(\underline{w})$.

(b) is clear from the definition of $\mathcal{V}_x^*$.

(c) follows from the definition of conditional probability.

(d) follows from the fact that averaged over all possible functions f_i, the conditional events shown in the equation are independent for different i's in $\mathcal{V}_x^*$.

Now we bound the expression given in (4.14) for any node $i \in \mathcal{V}_x^*$. Note that since $(w^{(i)}, Y_i^n(\underline{w})) \neq (w_0^{(i)}, Y_i^n(\underline{w}_0))$ at node i, $X_i^n(\underline{w}) = f_i(w^{(i)}, Y_i^n(\underline{w}))$ and $X_i^n(\underline{w}_0) =$

$f_i(w_0^{(i)}, Y_i^n(\underline{w}_0))$ are chosen independently and uniformly from $\{0,1\}^n$. Therefore the probability that they are the same in at least α_i specific locations is at most $2^{-\alpha_i}$. Looking at a fixed node i, $Y_{ij}^n(\underline{w}) = Y_{ij}^n(\underline{w}_0)$ for all j such that $(i,j) \in [\mathcal{V}_x, \mathcal{V}_x^c]$ only if all the locations that $X_i^n(\underline{w})$ and $X_i^n(\underline{w}_0)$ differ get erased on all these edges. Because of the δ-strong typicality of the erasure locations on edges $(i,j) \in [\mathcal{V}_x, \mathcal{V}_x^c]$, the number of locations at which erasure occurs on all the edges of interest, say $\alpha_i(\mathcal{V}_x)$, satisfies

$$\left| \frac{1}{n}\alpha_i(\mathcal{V}_x) - \Pr(Z_{ij} = 1,\ j : (i,j) \in [\mathcal{V}_x, \mathcal{V}_x^c]) \right| \leq \frac{\delta}{2^{|\{j:(i,j)\in[\mathcal{V}_x,\mathcal{V}_x^c]\}|}} \leq \delta.$$

Therefore $X_i^n(\underline{w})$ and $X_i^n(\underline{w}_0)$ cannot differ in more than $n(\Pr(Z_{ij} = 1,\ j : (i,j) \in [\mathcal{V}_x, \mathcal{V}_x^c]) + \delta)$ locations and the probability of this event is no more than

$$\exp\left(-n(1 - \Pr(Z_{ij} = 1,\ j : (i,j) \in [\mathcal{V}_x, \mathcal{V}_x^c]) - \delta)\right) = \exp\left(-n(1 - \prod_{j:(i,j)\in[\mathcal{V}_x,\mathcal{V}_x^c]} \epsilon_{ij} - \delta)\right). \tag{4.15}$$

Combining this with the last equation of (4.14) gives [11]

$$\begin{aligned}
\Pr(E(\underline{w})|\mathcal{A}_\delta^{(n)}) &\leq \sum_{\mathcal{V}_x:\mathcal{M}(\underline{w})\subset\mathcal{V}_x} \prod_{i\in\mathcal{V}_x^*} \exp\left(-n(1 - \prod_{j:\,(i,j)\in[\mathcal{V}_x,\mathcal{V}_x^c]} \epsilon_{ij} - \delta)\right) \\
&= \sum_{\mathcal{V}_x:\mathcal{M}(\underline{w})\in\mathcal{V}_x, d\notin\mathcal{V}_x} 2^{n|\mathcal{V}_x^*|\delta} \cdot \exp\left(-n\sum_{i\in\mathcal{V}_x^*}(1 - \prod_{j:\,(i,j)\in[\mathcal{V}_x,\mathcal{V}_x^c]} \epsilon_{ij})\right) \\
&\leq 2^{n|\mathcal{V}|\delta} \sum_{\mathcal{V}_x:\mathcal{M}(\underline{w})\subset\mathcal{V}_x, d\notin\mathcal{V}_x} 2^{-nC(\mathcal{V}_x)}.
\end{aligned} \tag{4.16}$$

[11] Using (4.15) it can be easily verified that the arguments that follow will exactly go through for correlated erasure events with cut-capacity, $C(\mathcal{V}_x)$, defined as in 4.22.

Combining (4.11) and (4.16) together gives

$$\begin{aligned}
\Pr(err) &\leq |\mathcal{V}|\delta + 2^{n|\mathcal{V}|\delta} \sum_{\underline{w}\in\underline{\mathcal{W}}-\{\underline{w}_0\}} \sum_{\substack{\mathcal{V}_x:\mathcal{M}(\underline{w})\subset\mathcal{V}_x \\ d\notin\mathcal{V}_x}} 2^{-nC(\mathcal{V}_x)} \\
&= |\mathcal{V}|\delta + 2^{n|\mathcal{V}|\delta} \sum_{\mathcal{M}\subset\mathcal{S}} \sum_{\substack{\underline{w}\in\underline{\mathcal{W}}-\{\underline{w}_0\} \\ \mathcal{M}(\underline{w})=\mathcal{M}}} \sum_{\substack{\mathcal{V}_x:\mathcal{M}(\underline{w})\subset\mathcal{V}_x \\ d\notin\mathcal{V}_x}} 2^{-nC(\mathcal{V}_x)} \\
&= |\mathcal{V}|\delta + 2^{n|\mathcal{V}|\delta} \sum_{\mathcal{M}\subset\mathcal{S}} \sum_{\substack{\mathcal{V}_x:\mathcal{M}\subset\mathcal{V}_x \\ d\notin\mathcal{V}_x}} \sum_{\substack{\underline{w}\in\underline{\mathcal{W}}-\{\underline{w}_0\} \\ \mathcal{M}(\underline{w})=\mathcal{M}}} 2^{-nC(\mathcal{V}_x)} \\
&= |\mathcal{V}|\delta + 2^{n|\mathcal{V}|\delta} \sum_{\mathcal{M}\subset\mathcal{S}} \sum_{\substack{\mathcal{V}_x:\mathcal{M}\subset\mathcal{V}_x \\ d\notin\mathcal{V}_x}} \prod_{s\in\mathcal{M}} (\lceil 2^{nR_s}\rceil - 1) 2^{-nC(\mathcal{V}_x)} \\
&\overset{(a)}{\leq} |\mathcal{V}|\delta + 2^{n|\mathcal{V}|\delta} \sum_{\mathcal{M}\subset\mathcal{S}} \sum_{\substack{\mathcal{V}_x:\mathcal{M}\subset\mathcal{V}_x \\ d\notin\mathcal{V}_x}} 2^{-n(C(\mathcal{V}_x)-\sum_{s\in\mathcal{M}} R_s)} \\
&\overset{(b)}{=} |\mathcal{V}|\delta + 2^{n|\mathcal{V}|\delta} \sum_{\mathcal{V}_x:\mathcal{V}_x\subset\mathcal{V}-\{d\}} \sum_{\mathcal{M}\subset\mathcal{S}\cap\mathcal{V}_x} 2^{-n(C(\mathcal{V}_x)-\sum_{s\in\mathcal{M}} R_s)} \\
&\overset{(c)}{\leq} |\mathcal{V}|\delta + 2^{n|\mathcal{V}|\delta} \sum_{\mathcal{V}_x:\mathcal{V}_x\subset\mathcal{V}-\{d\}} 2^{|\mathcal{V}_x\cap\mathcal{S}|} 2^{-n(C(\mathcal{V}_x)-\sum_{s\in\mathcal{S}\cap\mathcal{V}_x} R_s)}, \qquad (4.17)
\end{aligned}$$

where we have used the inequality $\lceil 2^{nR_s}\rceil - 1 \leq 2^{nR_s}$ in (a). Also (b) is derived by changing the order of summation and (c) follows from bounding $\sum_{s\in\mathcal{M}} R_s$ by $\sum_{s\in\mathcal{V}_x\cap\mathcal{S}} R_s$ in (c). Now by assumption the rate vector $(R_s, \quad s \in \mathcal{S})$ is inside the capacity region given in Theorem 4.2. Therefore for any partition of the nodes into $\mathcal{V}_x$ and $\mathcal{V}_x^c \ni d$ we have $C(\mathcal{V}_x) - \sum_{s\in\mathcal{S}\cap\mathcal{V}_x} R_s > 0$. Therefore the exponent in the last term of the above summation is negative. The above result holds for any $\delta > 0$ and n sufficiently large. By letting $n \to \infty$ and $\delta \to 0$, we can make the upper bound on the probability of error arbitrarily close to zero. Now by standard coding arguments we conclude that there exists some deterministic choice of encoding functions that has arbitrarily small probability of error for the rates in the achievable rate region $C(\mathcal{G}, \mathcal{S}, d)$.

4.6.2 Proof of Theorem 4.3

In this section we outline the proof of Theorem 4.3. The analysis is very similar to Theorem 4.1. The converse part is straightforward. We know that the sources can be recovered at all the destinations, therefore we have the same argument as the converse part of Theorem 1 for the sources and any of the destinations. In particular, for any destination $d_i, \quad i \in \mathcal{D}$, we have $(R_s, \quad s \in \mathcal{S}) \in C(\mathcal{G}, \mathcal{S}, d_i)$ Therefore any achievable rate vector should be in the intersection of these capacity regions, i.e,

$$(R_s,\ s \in \mathcal{S}) \in \cap_{d_i \in \mathcal{D}} C(\mathcal{G}, \mathcal{S}, d_i) = C(\mathcal{G}, \mathcal{S}, \mathcal{D}).$$

Hence the converse part is done.

In order to prove the achievability of the above rates, we can use the random coding argument of Section 1. Note that averaged over all the codebooks and functions, the probability of error for each destination goes to zero. Therefore using the union bound on probability of events, the probability of having an error in at least one destination (averaged over all the functions and codebooks) goes to zero. Using standard arguments, there exists some deterministic choice of codebooks and functions for which the probability of error in the network become arbitrarily small and that shows the achievability of the rates in $C(\mathcal{G}, \mathcal{S}, \mathcal{D})$ of Theorem 4.3 for the multiple destination case.

4.7 Linear Encoding

In Section 4.6.1.2 we showed the achievability of the capacity region as defined in Theorem 4.2 by using general random coding functions at different nodes of the network. In this section we restrict our attention to linear encoding schemes. The advantage of using a linear encoding scheme is that the decoding process becomes much easier. In this case, the equivalent transfer function of the network from any source to any destination, having the erasure locations at that destination, is linear. Hence, decoding at the destination is simply forming and solving a linear system of

equations.

In this section we show that linear encoders achieve capacity. Let us first define the linear block coding scheme with block length of n:

Recall that $\mathcal{W}^{(s)} = \{1, 2, \ldots, \lceil 2^{nR_s} \rceil\}$ is the message set for information source $s \in \mathcal{S}$. We assume that different messages are equiprobable and independent of each other. For any $w^{(s)} \in \mathcal{W}^{(s)}$, let $\underline{b(w^{(s)})}$ be the length-nR_s binary expansion of $w^{(s)} - 1$.

The encoding operation is as follows:

Each node $i \in \mathcal{V}$ transmits n linear combinations of the non-erased symbols received from its incoming edges and the binary representation of the message it wants to transmit across the network. More precisely, node i generates a random binary matrix B_i of size $n \times n(d_I(i) + R_i)$ where $d_I(i)$ is the in-degree of node i and R_i is the rate of the codebook used at node i (in the case where i is not a source of information $R_i = 0$). Each element of B_i is drawn i.i.d. Bernoulli(1/2). For a given sequence y, let $\widetilde{y}$ be a sequence derived by replacing every e with 0. Note that $\widetilde{y}$ and y have the same length.[12] If node i receives $Y_i^n = y_i^n$ on its incoming edges and wants to transmit message $w^{(i)}$ then it sends out $x_i = B_i \cdot [\underline{b(w^{(i)})}, \widetilde{y_i^n}]^\dagger$. (Since the input-output relation at each node is linear, setting the erased symbols equal to zero is the same as finding linear combinations of only the non-erased bits.)

Each destination d knows all the matrices B_i and also the erasure locations Z^n on all the links across the network. Since each received and transmitted symbol at any node is a linear combination of the elements of vector $b(\underline{w}) \stackrel{\triangle}{=} (\underline{b(w^{(s)})},\ s \in \mathcal{S})$. Therefore each destination receives a collection of linear combinations of elements of $b(\underline{w})$. Using $\{B_i\}_{i \in \mathcal{V}}$ and Z^n, destination node d can construct the matrix that corresponds to the linear input-output relation of the network. We denote this matrix by $F(\{B_i\}, Z^n)$, giving $\widetilde{Y_d^n}(\underline{w}) = F(\{B_i\}, Z^n) \cdot b(\underline{w})^\dagger$. Note that matrix F is a function of different nodes' encoding matrices $\{B_i\}$ and Z^n.

Now, upon receiving $Y_d^n = y \in \{0, 1, e\}^{nd_I(d)}$, the destination node d looks (solves) for the message vector $\underline{w} \in \underline{\mathcal{W}} \stackrel{\triangle}{=} \prod_{s \in \mathcal{S}} \mathcal{W}^{(s)}$ such that $F(\{M_i\}, Z^n) \cdot b(\underline{w})^\dagger = \widetilde{y}$. If

[12] The corresponding mapping from alphabet GF$(q) \cup \{e\}$ to GF(q) again replaces e with 0. This variation is useful for packet erasure networks.

there is a unique $\underline{w}$ with this property, node d declares it as the transmitted message vector, otherwise it declares an error. Note that the actual transmitted message vector, say $\underline{w}_0 \in \underline{\mathcal{W}}$, always satisfies the above property. Therefore, an error occurs only if there is another message vector $\underline{w} \neq \underline{w}_0$ such that $Y_d^n(\underline{w}) = Y_d^n(\underline{w}_0) = y$.

4.7.1 Achievability Result for Linear Encoding

Looking at the achievablity proof and probability of error analysis for general random coding in Sections 4.6.1.2 and 4.6.1.3, it can be easily verified that the linear case requires the same error events (4.10). Since the erasure vector Z^n is available at the destination, there is no difference between $\widetilde{Y}_i$ and Y_i and we can determine one from the other. By expanding the conditional error event $E(\underline{w})$ given $A_\delta^{(n)}$ for different cuts in the network, all of the relations up to step (d) of equation (4.14) go through for the linear case. In fact the relations up to step (d) only require the independence of encoding functions for different nodes of the network, which holds for the linear case. Now we look at the following probability in (4.14)

$$P_i \stackrel{\triangle}{=} \Pr\left(\bigcap_{j:(i,j)\in[\mathcal{V}_x,\mathcal{V}_x^c]} \{Y_{ij}^n(\underline{w}) = Y_{ij}^n(\underline{w}_0)\} \middle| \mathcal{A}_\delta^{(n)},\ \{(w^{(i)}, Y_i^n(\underline{w})) \neq (w_0^{(i)}, Y_i^n(\underline{w}_0))\} \right). \tag{4.18}$$

As in the general random coding argument, for a fixed i we have $Y_{ij}^n(\underline{w}) = Y_{ij}^n(\underline{w}_0)$ for all j such that $(i,j) \in [\mathcal{V}_x, \mathcal{V}_x^c]$, only if $X_i^n(\underline{w})$ and $X_i^n(\underline{w}_0)$ differ only in locations where an erasure occurs on all the edges of the interest. Because of strong typicality, the number of these location is at most $n(\prod_{j:\ \ (i,j)\in[\mathcal{V}_x,\mathcal{V}_x^c]} \epsilon_{ij} + \delta)$. Therefore $X_i^n(\underline{w})$ and $X_i^n(\underline{w}_0)$ should be the same in at least $n(1 - \prod_{j:\ \ (i,j)\in[\mathcal{V}_s,\mathcal{V}_s^c]} \epsilon_{ij} - \delta)$ locations. But by our encoding scheme this means that

$$B_i \cdot \underbrace{([w^{(i)}, Y_i^n(\underline{w})]^\dagger - [w_0^{(i)}, Y_i^n(\underline{w}_0)]^\dagger)}_{\underline{z}}$$

should be zero in at least $n(1 - \prod_{j:\ \ (i,j)\in[\mathcal{V}_s,\mathcal{V}_s^c]} \epsilon_{ij} + \delta)$ specific locations. Also note that since $(w^{(i)}, Y_i^n(\underline{w})) \neq (w_0^{(i)}, Y_i^n(\underline{w}_0))$, $\underline{z}$ is a non-zero vector. From the above

argument we have

$$\begin{aligned} P_i &\leq \Pr\left(B_i \cdot \underline{z} \text{ be 0 in at least } n\alpha_i \text{ specific locations} \,\middle|\, \underline{z} \neq 0\right) \\ &\overset{(a)}{\leq} 2^{-n\alpha_i} = 2^{-n(1-\prod_{j:\,(i,j)\in[\mathcal{V}_x,\mathcal{V}_x^c]}\epsilon_{ij}-\delta)}, \end{aligned} \tag{4.19}$$

where $\alpha_i = 1 - \prod_{j:\,(i,j)\in[\mathcal{V}_x,\mathcal{V}_x^c]} \epsilon_{ij} - \delta$ and (a) follows from the following lemma and its corollary. Proof of this lemma is provided in Appendix 4.9.2.

Lemma 4.6. *Let X be a non-zero vector of size $n \times 1$ from some finite field $GF(q)$. Suppose that A is a random matrix of size $m \times n$ with i.i.d. components distributed uniformly over $GF(q)$. Then the coordinates of $Y = A \cdot X$ are i.i.d. uniform random variables over $GF(q)$.*

Corollary 4.7. *The probability that $Y = A \cdot X$ is zero in k specific coordinates equals q^{-k}.*

Now note that by replacing P_i in (4.18) and (4.14) with its bound from (4.19) we get the same bound as (4.16) for random linear codes. Therefore linear operations are sufficient for achieving the capacity.

The main advantage of this is that the decoding operation can be carried out without exhaustive search of the exponential-sized codebook. The destination(s) only has to solve a system of linear equations, which can be done in polynomial time. This allows for faster and more efficient network operation.

4.8 Conclusions

We have obtained the capacity for a class of wireless erasure networks with broadcast and no interference at reception. We have generalized some of the capacity results that hold for wireline networks [4, 58] to these networks. Furthermore, we have shown that linear encoding suffices to achieve the optimal performance. We see from the proof that it is not necessary to perform channel coding and network coding separately from each other. In fact in [19, 62] we show that decoding at the relay nodes and operating

below the capacities of each link can actually significantly reduce the achievable rate. Therefore, unlike the wireline scenario where each link is made error-free by channel coding, and network coding is then employed on top of that, our scheme only requires a single encoding function. Only the destination has to decode the received signal.

Many problems related to wireless networks remain open. Generalizing the results in this chapter for other network problems is one possible extension of this work. As a first step, in [2], the problem of a single source wanting to send the same information to several destinations is considered. For these problems it can be shown that unlike wireline networks, the capacity region is not given by min-cut bounds. It is shown in [24] that the capacity region of multiple input erasure broadcast channels is given by time-sharing between users at different inputs. This result gives tighter outer-bounds on the capacity region of broadcast problems in erasure wireless networks.

It will also be interesting to see if similar results can be obtained for other types of networks, such as erasure wireless networks in which interference is incorporated in the reception model, networks involving channels other than erasure channels, etc.

4.9 Appendix

4.9.1 Proof of Converse

We have to show that any sequence of $(\lceil 2^{nR_1} \rceil, \ldots, \lceil 2^{nR_{|\mathcal{S}|}} \rceil, n)$ codes with $P_{d_1}^{(n)(s)} \to 0$ satisfies the bounds given in Theorem 2 (and Theorem 1). Let $\underline{W} = (W^{(s)}, \quad s \in \mathcal{S})$ be a random vector drawn i.i.d. from a uniform distribution over the set of message indices $\underline{\mathcal{W}}$. Let Z^n be the random vector describing the erasure locations, i.e., $Z^n =$

$(Z_{ij,t},\ (i,j)\in\mathcal{E}, t\in\{1,\ldots,n\})$. Consider an s-d cut given by s-set $\mathcal{V}_s$. We have,

$$
\begin{aligned}
& n\sum_{s\in\mathcal{V}_d^c\cap\mathcal{S}} R_s \\
= \ & H(W^{(s)}, \quad s\in\mathcal{S}\cap\mathcal{V}_d^c) \\
= \ & I((W^{(s)}, \quad s\in\mathcal{S}\cap\mathcal{V}_d^c); Y_d^n, Z^n) + H((W^{(s)}, \quad s\in\mathcal{S}\cap\mathcal{V}_d^c)|Y_d^n, Z^n) \\
\overset{(a)}{\leq} \ & I((W^{(s)}, \quad s\in\mathcal{S}\cap\mathcal{V}_d^c); Y_d^n, Z^n) + n\epsilon_n \\
\overset{(b)}{\leq} \ & I((W^{(s)}, \quad s\in\mathcal{S}\cap\mathcal{V}_d^c); Y^n(\mathcal{V}_d^c), Z^n) + n\epsilon_n \\
\overset{(c)}{=} \ & I((W^{(s)}, \quad s\in\mathcal{S}\cap\mathcal{V}_d^c); Y^n(\mathcal{V}_d^c)|Z^n) + n\epsilon_n \\
= \ & H(Y^n(\mathcal{V}_d^c)|Z^n) - H(Y^n(\mathcal{V}_d^c)|Z^n, (W^{(s)}, \quad s\in\mathcal{S}\cap\mathcal{V}_d^c)) + n\epsilon_n \\
\overset{(d)}{=} \ & H(Y^n(\mathcal{V}_d^c)|Z^n) + n\epsilon_n \\
\overset{(e)}{\leq} \ & H(Y^n(\mathcal{V}_d^c)|(Z_{ij}^n, \quad (i,j)\in[\mathcal{V}_d^c,\mathcal{V}_d])) + n\epsilon_n \\
\overset{(f)}{\leq} \ & \sum_{t=1}^{n}\sum_{i\in\mathcal{V}_d^{c*}} H(Y_{ij,t}, \quad j:(i,j)\in[\mathcal{V}_d^c,\mathcal{V}_d]|Z_{ij,t}, \quad j:(i,j)\in[\mathcal{V}_d^c,\mathcal{V}_d]) + n\epsilon_n \\
\overset{(g)}{\leq} \ & \sum_{t=1}^{n}\sum_{i\in\mathcal{V}_d^{c*}} H(Y_{ij,t}, \quad j:(i,j)\in[\mathcal{V}_d^c,\mathcal{V}_d]) - H(Z_{ij,t}, \quad j:(i,j)\in[\mathcal{V}_d^c,\mathcal{V}_d]) + n\epsilon_n \\
\overset{(h)}{=} \ & \sum_{t=1}^{n}\sum_{i\in\mathcal{V}_d^{c*}} H(Y_{ij,t}, \quad j:(i,j)\in[\mathcal{V}_d^c,\mathcal{V}_d]) - H(Y_{ij,t}, \quad j:(i,j)\in[\mathcal{V}_d^c,\mathcal{V}_d]|X_{i,t}) + n\epsilon_n \\
= \ & \sum_{t=1}^{n}\sum_{l\in\mathcal{V}_d^{c*}} I(X_{l,t}; (Y_{lj,t}, \quad j:(l,j)\in[\mathcal{V}_d^c,\mathcal{V}_d])) + n\epsilon_n \qquad (4.20) \\
\overset{(i)}{\leq} \ & \sum_{t=1}^{n}\sum_{l\in\mathcal{V}_d^{c*}} (1-\Pr(Z_{lj}=1, \quad j:\ (l,j)\in[\mathcal{V}_d^c,\mathcal{V}_d])) + n\epsilon_n \qquad (4.21) \\
\overset{(j)}{\leq} \ & \sum_{t=1}^{n}\sum_{l\in\mathcal{V}_d^{c*}} (1-\prod_{j:\,(l,j)\in[\mathcal{V}_d^c,\mathcal{V}_d]} \epsilon_{lj}) + n\epsilon_n \\
= \ & nC(\mathcal{V}_d^c) + n\epsilon_n
\end{aligned}
$$

where

(a) follows from Fano's inequality since message $W^{(s)}$ can be decoded at node d from Y_d^n and the erasure locations Z^n across the network.

(b) follows from the properties of the block code defined in Section 4.4 and data

processing inequality. The causality of the block code and also the deterministic structure of the relaying functions can be used to inductively show that $(W^{(s)}, \quad s \in \mathcal{S} \cap \mathcal{V}_d^c) - Y^n(\mathcal{V}_d^c) - Y_d^n$ forms a Markov chain for any cut. Applying the data processing inequality gives inequality (b).

(c) follows since messages and erasure locations are independent from each other.

(d) follows since the output of every channel is a deterministic function of the erasure locations Z^n and the transmitted messages $(W^{(s)}, \quad s \in \mathcal{V}_d^c \cap \mathcal{S})$. Therefore the second conditional entropy is zero.

(e) follows from the fact that conditioning reduces the entropy.

(f) follows from the fact that conditioning reduces the entropy and $H(X_1, \ldots, X_m) \leq \sum_{i=1}^m H(X_i)$ for any collection of random variables.

(g) follows from the fact that $(Z_{ij,t}, \quad j : (i,j) \in [\mathcal{V}_d^c, \mathcal{V}_d])$ is a deterministic function of $(Y_{ij,t}, \quad j : (i,j) \in [\mathcal{V}_d^c, \mathcal{V}_d])$.

(h) follows since $H((Y_{ij,t}, \quad j : (i,j) \in [\mathcal{V}_d^c, \mathcal{V}_d])|X_{i,t}) = H(Z_{ij,t}, \quad j : (i,j) \in [\mathcal{V}_d^c, \mathcal{V}_d])$.

(i) follows from the capacity of the memoryless erasure channel. Here the transmitter transmits $X_{l,t}$ and the receiver has access to $Y_{lj,t}\,, (l,j) \in [\mathcal{V}_d^c, \mathcal{V}_d]$. The receiver experiences an erasure only if all channels $(l,j) \in [\mathcal{V}_d^c, \mathcal{V}_d]$ simultaneously suffer an erasure. Therefore the equivalent channel's erasure probability is $\Pr(Z_{lj,t} = 1, \ j : (l,j) \in [\mathcal{V}_d^c, \mathcal{V}_d])$.

(j) follows since in the case of independent erasure events $\Pr(Z_{lj,t} = 1, \ j : (l,j) \in [\mathcal{V}_d^c, \mathcal{V}_d])$ is equal to $\prod_{j:(l,j)\in[\mathcal{V}_d^c,\mathcal{V}_d]} \epsilon_{lj}$

Remark 3. *As we observe from (4.20), the upper bound is in terms of the mutual information between the input and outputs of every broadcast system (i.e., a node and its outgoing edges) in the network. We should also mention that the same kind of*

upper bound of (4.20) holds for more general networks (not necessarily erasure) that have interference-free and broadcast property.

Remark 4. *From (4.21), we see that in the case of correlated erasure events, i.e., when Z_{ij}s are dependent (however still data-independent), we can find an upper bound for the maximum achievable rate for each cut. Furthermore, as mentioned in the footnote of page 20, it can be verified that the probability of error analysis of Section 4.6.1.3 is valid for the correlated erasure events with the following definition of the cut-capacity*

$$C(\mathcal{V}_s) = \sum_{i \in \mathcal{V}_s^*} \Big(1 - \Pr(Z_{ij} = 1,\ j : (i,j) \in [\mathcal{V}_s, \mathcal{V}_s^c]) \Big). \tag{4.22}$$

Therefore, the capacity results of this chapter go through for the correlated erasure events as well.

4.9.2 Proof of Lemma 4.6

First note that if x and y are independent uniform random variables over $GF(q)$ it can be easily verified that $x + y$ is also uniformly distributed over $GF(q)$. By a simple induction it is straightforward that sum of any number of independent uniform random variables is uniformly distributed. By assumption different rows of A are independent from each other. Also each element of Y is a linear combination of elements of one specific row. Therefore different elements of Y are independent from each other. Now we show that elements of Y are uniformly distributed. Without loss of generality look at first element, i.e.,

$$y_1 = \sum_{i=1}^{n} a_{1k} x_k.$$

Note that for non-zero x_ks, the $a_{1k}x_k$s are independent uniformly distributed random variables. Hence y_1 is a sum of a number of independent uniform random variables over $GF(q)$, and, based on the above discussion y_1 is uniformly distributed.

Remark: Using Bayes rule, we can easily check that if X is a non-zero uniform random vector over $GF(q)$ and it is independent of A, then $A \cdot X$ is a uniform random vector.

Chapter 5

A Practical Scheme for Wireless Network Operation

In the previous chapter, we obtained an exact capacity result for a certain class of networks called wireless erasure networks. The strategies used to achieve capacity included a random codebook and random encoding by the relay nodes. In addition, side-information regarding erasure locations and encoding functions is required at the destination(s). In many practical systems, it is not possible to use these strategies because of the complexity of computation and the delays involved. Therefore, in this chapter, we consider networks in which nodes are restricted to performing a fixed collection of very simple operations. We consider directed and acyclic networks with either Gaussian fading links or erasure links. In the former case, we incorporate broadcast and interference in the usual way, and in the latter case, the model is largely identical to that of the previous chapter.

On the face of it, the simplest operation that every node can do is try and decode the received message, possibly without error, and send it on. This would make every link or subnetwork act in an error-free manner. In fact, in many problems in wireline networks, it is known that this leads to optimal operation, that is, achieving capacity on each link or sub-network is optimal for the entire network operation. In this chapter, we will see that the same is not true for wirless networks. We present examples of wireless networks in which decoding and achieving capacity on certain links or sub-networks gives us lower rates than other simple schemes, like simply

forwarding the data, with erasures or noise. This implies that the separation of channel and network coding that holds for many classes of wireline networks does not, in general, hold for wireless networks.

We then consider the question of optimal operation of Gaussian and erasure wireless networks where nodes are permitted only two possible operations – nodes can either decode what they receive (and then re-encode and transmit the message) or simply forward it. We present a centralized greedy algorithm that returns the optimal scheme from the exponential-sized set of possible schemes. This algorithm will go over each node at most once to determine its operation and hence is very efficient. We also present a decentralized algorithm whose performance can approach the optimum arbitrarily closely in an iterative fashion. It is important to note that once nodes know the appropriate operation that they need to perform, the schemes do not require any channel side-information at the destination.

5.1 Introduction

In a wireline network with a single source and a single destination, we can think of information flow as fluid flow and obtain a max-flow min-cut result to get capacity. This treatment closely follows that of the Ford-Fulkerson [66] algorithm to give a neat capacity result. This has been well-understood for many years. However, until recently, similar max-flow min-cut capacity results were not known for any other class of network problems. Before we describe the recent results obtained in network problems, let us understand the general network problem. This can be stated in the context of a multi-terminal network [71] as follows. We have a set of nodes and the "channel" between these is specified by a probability transition function that governs the relationship between the signals transmitted by the nodes and signals received by the nodes. Every node can have messages that it wants to send to every other node. Because of the generality of this model, it can be tailored to describe many practical systems easily. For instance, several wireless as well as wireline systems, (stationary) ad hoc and sensor networks, etc., can be modeled by choosing a suitable probability

transition function.

In recent years large ad hoc networks have received a lot of attention, starting with the work of Gupta and Kumar [35]. Most results involving these networks use relaying as a tool and consider issues like throughput, power efficiency and distortion. In addition, cooperation is a technique that has been shown to be very effective [70]. However, these methods study asymptotically large networks and give scaling laws rather than exact results for the performance measures that they study. In fact, finding the exact capacity region in this general setting is extremely challenging. In [71] outer bounds on the capacity region can be found. These have the form of "min-cut" upper bounds. Such an upper bound formalizes the intuitively satisfying notion that the rate from node a to node b cannot exceed the rate that any cutset of edges from a to b can support. However, determining whether schemes of network operation that reach this upper bound exist or not has proved to be very difficult. Even in simple relay networks, i.e., networks having one source node, one destination node and a single other node (called the relay node), the answer to this question is not known in general [71]. Only in special cases of the probability transition function (defined as "degraded" distributions) do we know of schemes that can reach the upper bounds and thus attain capacity.

In this context, the results in [68, 64] are remarkable. They say that, in a wireline network setting, we can achieve the min-cut upper bounds for multicast problems where one source node sends the same message to several sink nodes. It turns out that using network coding techniques we can achieve the min-cut capacity of the network. Further, [58] put this problem in an algebraic framework and presented *linear* schemes that also achieved this capacity. In addition, for some more general multicast problems, capacity has been shown to be achievable using linear network coding [58]. The work of [74, 65, 55] demonstrates the strengths of this algebraic approach.

In all these capacity-achieving schemes, the min-cut upper bounds are reached through separate channel and network coding: each link in the wireline network can be made error-free by means of channel coding and then network coding can be

employed to determine which messages should be transmitted on each link. This is quite unexpected and leads us to wonder if such a separation is optimal in more general network settings.

In investigating this question, we first present simple *wireless* networks where this principle of separation fails. Thus, in these networks, it is suboptimal for each relay to decode prior to retransmission. This observation was first made in [62, 20]. We will also suggest some schemes of operation that will outperform those that require the ability of relay nodes to decode.

We focus attention on two wireless network models: Gaussian wireless networks and erasure wireless networks. The first model has Gaussian channels as links and incorporates broadcast as well as interference. The second model has erasure channels as links and incorporates broadcast but not interference. For these models, we show that making links error-free sometimes prohibits optimal performance. In fact, sometimes it is better for nodes to forward their data unchanged.

We propose a scheme of network operation that permits nodes only two operations. One is decoding to get the original data and then resending the same message as the source. The other is forwarding the data as received. Since each node has two options, we have an exponential-sized set of possible operations. We present an algorithm that goes over each node at most once to find the optimal operation among this set of restricted operations. This algorithm is greedy and optimal. We also present an algorithm that iteratively approaches the best rate. The algorithm is "decentralized": in each iteration each node needs only one bit of information from the destination and no knowledge of the rest of the network in order to determine its own operation.

The organization of this chapter is as follows. In Section 5.2 we present the two wireless network models. In Section 5.3 we show that with these wireless models, making links or sub-networks error-free can be sub-optimal. In Section 5.4 we formally state the two operations allowed at each node. We describe the full design problem in Section 5.5. In Section 5.6 we investigate the rate implications of the decode and forward strategies. will see how rates are calculated for all nodes in the network and how asking certain nodes to decode and others to forward can affect the rate

of the network. We describe our algorithm in Section 5.7 and prove its optimality in Section 5.8. Section 5.9 contains examples showing that the gap between the "all nodes decode" strategy and our method can be significant. In Section 5.10 we discuss the decentralized algorithm. We present upper bounds on the rate achievable by our scheme in Section 5.11. Conclusions and further questions are presented in Section 5.12.

5.2 Two Wireless Network Models

In this section we formalize two wireless network models. These are Gaussian networks and erasure networks. In both cases the network consists of a directed, acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of vertices and $\mathcal{E}$ is the set of directed edges where each edge is a communication channel. We will denote $|\mathcal{V}| = V$ and $|\mathcal{E}| = E$. Also, we will have $\mathcal{V} = \{v_1, \cdots, v_V\}$ and $\mathcal{E} = \{(v_i, v_j) : (v_i, v_j) \text{ is an edge}\}$. We will assume, without loss of generality, that $s = v_1$ is the source node and $d = v_V$ is the destination. The remaining nodes are the relay nodes that must aid communication between s and d. We will assume that every edge is on some directed path from s to d. If we have edges other than these, we remove them and what remains is our graph $\mathcal{G}$. We will denote the message transmitted by vertex v_i by $X(v_i)$ and that received by node v_j by $Y(v_j)$. Figure 5.1 represents a network with six vertices and nine edges

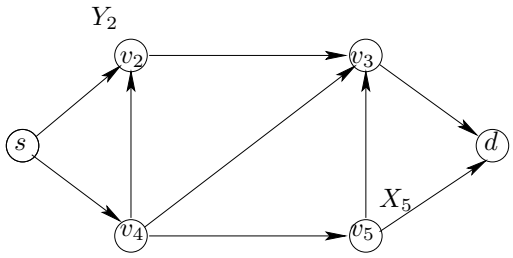


Figure 5.1: Example of a network with six vertices and nine edges. v_1 is the source s and v_6 is the destination d. $X(v_5)$ is the message transmitted by v_5 and $Y(v_2)$ is that received by v_2.

where v_1 is the source s and v_6 is the destination d. $X(v_5)$ is the message transmitted by v_5 and $Y(v_2)$ is that received by v_2.

(a) **Gaussian Wireless Networks:** In these networks, each edge (v_i, v_j) of the network is a Gaussian channel with some fixed attenuation factor $h_{i,j}$ associated with it. In a practical system, this may be some pathloss that depends on the physical distances between the nodes. We will assume $h_{i,j}$ to be a non-negative constant. We will assume that nodes broadcast messages, i.e., a node transmits the same message on all outgoing edges. Assuming that Figure 5.1 represents a Gaussian wireless network, $X(v_5)$ is the message transmitted on edges (v_5, v_3) and (v_5, v_6). We will also assume interference, i.e., the received signal at node v_i is the sum of all the signals transmitted on edges coming into it and additive white Gaussian noise n_i of variance σ_i^2. Therefore, in general, we have

$$Y(v_i) = n_i + \sum_{v_j:(v_j,v_i)\in\mathcal{E}} h_{j,i} X(v_j).$$

All n_is are assumed independent of each other as well as of the messages. For Figure 5.1 this implies that $Y(v_2) = h_{1,2}X(v_1) + h_{4,2}X(v_4) + n_2$. We will assume that all transmitting nodes have a power constraint of P.

(b) **Erasure Wireless Networks:** In these networks, each edge (v_i, v_j) of the network is a binary erasure channel with erasure probability $\epsilon_{i,j}$. In addition, we assume that nodes (other than the source node) can transmit erasures and they are received as erasures with probability one. Denoting erasure by $*$, this assumption means that edges can also take $*$ as input and this is always received as $*$. In short, the channel for edge (v_i, v_j) (for $v_i \neq s$) is modified as in Figure 5.2. We incorporate broadcast in the model, i.e., each transmitting node must send out the same signal on each outgoing edge. Now assuming that Figure 5.1 represents a wireless erasure network, v_5 transmits $X(v_5)$ on edges (v_5, v_3) and (v_5, v_6).

However, we do not permit interference. This means that a node having several incoming edges sees messages from each edge without their interfering with each other. In general, if v_i has $\gamma_I(i)$ incoming edges, it will see $\gamma_I(i)$ messages that

do not interfere with each other.[1] In Figure 5.1, we see that $Y(v_2)$ consists of two received messages – the message coming in on edge (v_1, v_2) (which is $X(v_1)$ with some bits erased) as well as the message coming in on edge (v_4, v_2) (which is $X(v_4)$ with some bits erased).

Finally, we mention that instead of the regular binary erasure channel, we can consider a channel with any finite alphabet $\mathcal{A}$ as the input alphabet and get a more general erasure wireless network model. Our results go through for this also, but for simplicity, we restrict ourselves to binary inputs.

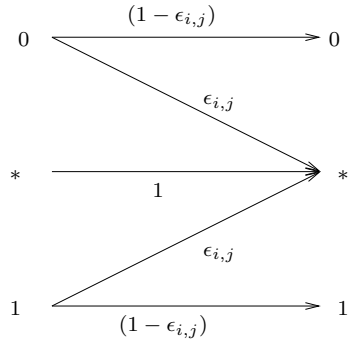


Figure 5.2: Modified erasure channel. We allow erasures to be transmitted as well as the bits 0 and 1. Erasures are always received as erasures.

For both networks, we will assume instantaneous transmission on all links.

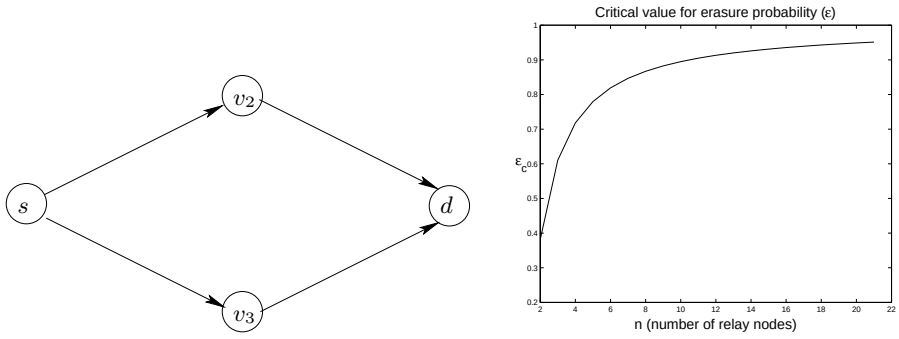
5.3 Optimizing over Sub-networks does not work

Theorem 5.1. *For the wireless networks described in Section 5.2, making sub-networks error-free can be suboptimal.*

Proof. We give some examples to demonstrate this.

- **Gaussian Relay Networks**: Consider a Gaussian parallel relay network consisting of two relay nodes and one source-destination pair. See Figure 5.3(a).

[1] There exist network models in the physical layer that incorporate interference, which, when abstracted to an erasure network model act similarly to the interference-free model we have described here. For instance, simple division multiple access schemes, such as TDMA, FDMA or CDMA can be used to eliminate the interference.



(a) Graph representation of a relay network with two relay nodes.

(b) Critical value of erasure probability for k relay nodes.

Figure 5.3: Proof of Theorem 5.1. We see that for certain erasure probabilities, having the relay nodes decode causes the rate to the destination to decrease. Thus, making the subnetwork $\{s, v_2, v_3\}$ error-free can be suboptimal.

All four channel coefficients are assumed to be one. The relay nodes v_2 and v_3 are solely to aid communication from source to destination. We assume that the noise power at each receiver is σ^2 and the transmit power at each node is P. Let $\rho \triangleq \frac{P}{\sigma^2}$ be the signal-to-noise-ratio (SNR).

One way to view the network is as a cascade of a broadcast channel (from s to $\{v_2, v_3\}$) and a multiple access channel (from $\{v_2, v_3\}$ to d). This is equivalent to assuming that the relays decode their messages correctly and code them again and transmit. If the relays are receiving independent information at rates R_1 and R_2, we have $R_1 + R_2 \leq \log(1+\rho)$ as the capacity region. These rate pairs (R_1, R_2) can be supported by the multiple access channel and hence the maximum rate from s to d is no greater than $\log(1+\rho)$. If the relays are receiving exactly the same information from the source, the maximum rate of this is $\log(1+\rho)$. In this case, the multiple access channel is used for correlated information and can support rates up to $\log(1+4\rho)$. In either case, asking the relay nodes to decode limits the rate from s to d to $\log(1+\rho)$. (We note also that the broadcast sub-network is the bottleneck in both cases.)

Now consider another strategy in which the relay nodes do not decode but only normalize their received signal to meet the power constraint and transmit it to

the destination. In this case the received signal at the destination is

$$Y(v_4) = \sqrt{\frac{P}{P+\sigma^2}}(2X(v_1) + n_2 + n_3) + n_4$$

where $X(v_1)$, $Y(v_4)$, n_2, n_3, n_4 are, respectively, the transmitted signal from the source, the received signal at the destination and the noises introduced at v_2, v_3 and d. Thus, the signal received by d is a scaled version of $X(v_1)$ with additive Gaussian noise. The maximum achievable rate, denoted by R_f, is

$$R_f = \log\left(1 + \frac{\frac{4P^2}{P+\sigma^2}}{\sigma^2 + \frac{2P\sigma^2}{P+\sigma^2}}\right) = \log\left(1 + \frac{4\rho^2}{3\rho+1}\right)$$

where ρ is as before. Here, the subscript f stands for *forwarding*.

Comparing R_d and R_f, we can see $\rho = 1$ is a critical value in the following sense. For $\rho > 1$, we have superior performance in the forwarding scheme and for $\rho < 1$ we have better rate with relay nodes decoding and re-encoding. This implies that making a sub-network error-free (in this case, the broadcast section, or the links (v_1, v_2) and (v_1, v_3)) can sometimes be sub-optimal.

We note that decoding at one of the relay nodes and forwarding at the other is always sub-optimal.

In general, if we have $k(\geq 2)$ relay nodes in parallel rather than two, it can be easily checked that

$$R_d = \log(1+\rho) \quad \text{and} \quad R_f = \log\left(1 + \frac{k^2\rho^2}{(k+1)\rho+1}\right).$$

With this we get a critical value of $\rho = \frac{1}{k^2-k-1}$ below which decoding is better and above which forwarding is better. Clearly, this goes to zero for large k. Therefore in the limit of $k \to \infty$ it is always favorable to forward.

It turns out that this fact is also true for Gaussian relay networks in the presence of fading. The work of [62] shows that for fading Gaussian relay networks with n nodes, the asymptotic capacity achievable with the relay nodes decoding (and

re-encoding) scales like $O(\log \log n)$ whereas with the forward scheme it scales like $O(\log n)$.

Similar problems are considered in [59] and [60]. The former considers bounds and achievable rates for the Gaussian network with two parallel links and the latter considers a network with a single source and destination and the other nodes acting as relays. The second result shows that the maximum rate achievable is $O(\log n)$. This is the same as that achieved by forwarding in our scheme.

- **Erasure Relay Network**: Consider, once again, the network of Figure 5.3(a), where, now, each link represents an erasure channel with erasure probability $\epsilon_{i,j} = e$. Since we have broadcast, node s transmits the same messages to relay nodes v_2 and v_3. If the relay nodes decode and re-encode, the rate is bounded by the sum-rate capacity of the broadcast system, which gives

$$R_d = 1 - e.$$

If the relay nodes simply forward what they receive, it is easy to see that the destination sees an effective erasure probability of $(1-(1-e)^2)$. (We will spell out how to do this calculation for a general network in Section 5.6.) Forwarding erasures is possible since we are assuming the modified erasure channel of Figure 5.2. With this we have $R_f = 1-(1-(1-e)^2)^2$. Comparing R_f and R_d, we can see that $e = \frac{3-\sqrt{5}}{2}$ is a critical value, above which decoding and re-encoding is better and below which forwarding is better.

Thus we see that for this network also, making the broadcast sub-network error-free is not always optimal.

In general, if we have k relay nodes in parallel rather than two, we have

$$R_d = 1 - e \quad \text{and} \quad R_f = 1 - (1 - (1-e)^2)^k$$

and the critical value of e is as plotted in Figure 5.3(b). Below this, forwarding

is better and above this decoding is better. In the limit of large k, it is always better to forward.

From this we see that making links or sub-networks error-free does not ensure optimal network operation. It can sometimes be provably sub-optimal. □

In this proof a simple operation like forwarding the received data proved to be better than decoding it. We understand this as follows. Because of the broadcast present in wireless networks, the same data naturally gets passed on to the destination along many different paths. Therefore some nodes receive better versions of the data on incoming links than other nodes and are automatically in a better position to decode. Forcing all the nodes to decode and be error-free only imposes additional bottlenecks on the rate. Therefore it is beneficial to carefully check the quality of the effective signal that various nodes get to see and then decide whether to ask them to decode or not.

5.4 A Possible Set of Network Operations

It follows from the previous discussions that to obtain the optimum rate over wireless networks, the nodes must perform operations other than just decoding. Determining what the optimum operation at each node should be, especially for a general wireless network, appears to be a daunting task. We shall therefore simplify the problem by allowing one of only two operations at every node. One will be the decode and re-encode operation as before. The other is the far simpler operation of forwarding the received data as is. The first operation, viz., decode and re-encode, is typically the only operation used in multihop networks and many wireline networks. In effect, we are attempting to attain higher rates by introducing the additional operation of forwarding.

We will assume that the network operates in blocks of length n. We assume that the source s has a set of message indices

$$\Omega = \{1, 2, \ldots, 2^{\lfloor nR \rfloor}\}$$

and an encoding function

$$f : \Omega \to \mathcal{X}^n$$

where $\mathcal{X}$ is $\mathbb{R}$ for the Gaussian wireless network and $\{0, 1\}$ for the erasure wireless network. To transmit message $i \in \Omega$, the source transmits $f(i)$. With this the source operates at rate R. $\{f(1), f(2), \ldots, f(2^{\lfloor nR \rfloor})\}$ is the set of codewords or possible transmitted messages. This set is called the codebook and is denoted by $\mathcal{C}$. We assume that all nodes have the codebook. For the Gaussian network we will assume that the codebook meets the power constraint, i.e., $E\|f(i)\|^2 \leq P$.

In this chapter, we restrict the relay nodes to two operations. These have been introduced in the examples of Section 5.3, viz., "forward" and "decode and re-encode." We now state them formally.

(a) **Decode and Re-encode**: This operation implies that when node v_i receives message $Y(v_i)$ it performs ML decoding of $Y(v_i)$ to determine which message index was transmitted by s. Since it has the codebook, it re-encodes the message using the same codeword that the source s would have used and transmits the same codeword. In short, it should act like a copy of the source.

However, for this to happen, we need the decoding to be error-free. This implies that the rate R at which the source operates should be no greater that the maximum rate at which node v_i can decode. We will see the relevance of this constraint in Section 5.5.

(b) **Forward**: We will describe this operation separately for the two network models. In the Gaussian network, node v_i receives message $Y(v_i)$ given by

$$Y(v_i) = n_i + \sum_{v_j:(v_j,v_i)\in\mathcal{E}} h_{j,i} X(v_j). \tag{5.1}$$

"Forwarding" implies that the node normalizes this signal to meet the power constraint and then transmits the message. Therefore it transmits $X(v_i)$ given

by

$$X(v_i) = \sqrt{\frac{P}{E\|Y(v_i)\|^2}}\ \ Y(v_i).$$

We will assume that $E\|Y(v_i)\|^2$ is known to v_i.

For the erasure network, nodes either decode without error and transmit the original codeword or "forward" the received data. Consider node v_i which sees data coming in on several edges, in the form of n-length blocks of bits and erasures. For the bth bit of such a block, it either sees erasures on every edge (and this sees an *effective* erasure) or gets to see the bit on at least one incoming edge. (It cannot happen that the node sees 1 on a particular edge and 0 on another edge for the bth position. This is because of our assumption that whenever an earlier node decodes, it does so without error.) Therefore in our interference free model, every relay node sees an effective erasure channel from the source, i.e., it sees the codeword transmitted by the source with some bits erased. "Forwarding" means broadcasting this sequence of bits and erasures.

Note that the effective erasure probability seen by node v_i is a function of the network topology and parameters, $\epsilon_{i,j}$. We will see in Section 5.6.3 how this effective erasure probability can be calculated.

By restricting ourselves to only two operations, we have ensured that all nodes in the network see a Gaussian channel (with some effective SNR) or erasure channel (with some effective erasure probability) with respect to the transmitted codeword. Therefore, they can do ML decoding or typical set decoding if R is no greater than the rate that they can support. We will always ensure that R satisfies this constraint.

We can think of both operations as specific forms of network coding. In both networks and with both operations, all the information coming in at a node on different edges gets pooled together – this happens automatically in the Gaussian network and is done by the node itself in the erasure network. But the node has the choice of trying to decode, thus imposing a rate constraint, or simply forwarding the information, hoping that some other node would have a better chance of decoding.

Having described the two operations permitted to the relay nodes in the two networks, we are now ready to formally state the problem.

5.5 Problem Statement

Since we allow only two operations to nodes, viz., "decode and re-encode" and "forward," and every relay node must perform one of these, it is enough to specify the set of relay nodes that "decode and re-encode" in order to completely specify the working of the network. The source and destination will always be excluded from this set.

If a set $D \subseteq \mathcal{V} - \{s, d\}$ is the set of nodes that "decode and re-encode," we will call D a **policy** for network operation.

Under policy D, each node of the network sees an effective (Gaussian or erasure) channel from the source. Let the effective SNR that node v_i sees under policy D be denoted by $\rho_D(v_i)$ for Gaussian networks. For erasure networks we denote the effective erasure probability seen by node v_i under policy D by $e_D(v_i)$. Therefore the rate that node v_i can support under policy D is $\log(1 + \rho_D(v_i))$ or $(1 - e_D(v_i))$ for Gaussian or erasure networks, respectively. In general we will call this $R_D(v_i)$. Nodes in D as well as the destination must be able to perform error-free decoding. This means that the rate at which the source transmits must be no greater than the rates at which these nodes can decode. This tells us that under policy D, the rate R at which we can operate the network is constrained by

$$R \leq \min_{v_i \in D \cup \{d\}} R_D(v_i). \tag{5.2}$$

We denote this minimum by R_D.

$$R_D = \min_{v_i \in D \cup \{d\}} R_D(v_i). \tag{5.3}$$

Intuitively, asking some nodes to decode means that there are more copies of the source in the network and hence the rate that the destination can support increases. On the other hand, asking a node to decode introduces a constraint on the rate R.

This is the tradeoff for any policy D. For instance, in Figure 5.1 consider nodes v_2 and v_4. If v_4 forwards, node v_2 sees an effective erasure probability of $\epsilon_{4,2}\epsilon_{1,2} + \epsilon_{1,4}\epsilon_{1,2}(1-\epsilon_{4,2})$. (We will see how this has been calculated in Section 5.6.3.) On the other hand, if v_4 decodes, node v_2 is at an advantage since it sees a lower effective erasure probability, viz., $\epsilon_{1,2}\epsilon_{4,2}$. However, asking v_4 to decode puts a constraint on the rate as seen by (5.2) since the rate that v_4 can support is only $(1-\epsilon_{1,4})$. This constraint is $R_D \leq 1-\epsilon_{1,4}$.

Our problem is to find the policy that gives the best rate, i.e., to find D such that R_D is maximized, viz.,

$$\max_{D} \min_{v_i \in D \cup \{d\}} R_D(v_i).$$

First we need to address the question of finding $R_D(v_i)$, i.e., of finding the rate at node v_i under policy D. Recall that $X(v_i)$ and $Y(v_i)$ are the transmitted and received messages at node v_i. If we are using policy D, we will denote these by $X_D(v_i)$ and $Y_D(v_i)$. We may drop the subscript D if it is clear which policy we are referring to. Note that for the source, the transmitted message is $X(v_1)$ irrespective of the policy.

5.6 Determining the Rate at a Node – $R_D(v_i)$

In this section we describe a method to find the rate at an arbitrary node v_i when the set of decoding nodes is given by D. Therefore, we need to find the effective SNR or erasure probability of the received signal $Y_D(v_i)$. In order to do that, we need the concept of a partial ordering on the nodes.

5.6.1 Partial Ordering of Nodes

Consider two distinct nodes v_i and v_j of the network. Exactly one of the following three will occur:

(a) There is a directed path from v_i to v_j. In this case we will say that $v_i < v_j$.

(b) There is a directed path from v_j to v_i. In this case we will say that $v_j < v_i$.

(c) There is no directed path from v_i to v_j or from v_j to v_i. In this case we will say that v_j and v_i are incomparable.

Note that since we assume acyclic networks, we cannot have directed paths both from v_i to v_j and from v_j to v_i. Thus we have a partial ordering for nodes in the network. For example, in Figure 5.1, we have $v_4 < v_3$ but v_2 and v_5 are incomparable. Note that the partial ordering gives us a (non-unique) sequence of nodes starting with s such that for every v_i, all the nodes v_j that satisfy $v_j < v_i$ are before it in the sequence [72]. Call such a sequence $\mathcal{S}$. A possible sequence $\mathcal{S}$ for Figure 5.1 is $(s, v_4, v_2, v_5, v_3, d)$.

Next we address the issue of determining the rate under a particular policy. We discuss this separately for Gaussian wireless networks and erasure wireless networks.

5.6.2 Finding the Rate in Gaussian Wireless Networks

Recall that $Y_D(v_j)$ is the received signal at v_j under policy D. Once we know $Y_D(v_j)$ we can determine the signal power and the noise power in it. Denote these by $P_D(v_j)$ and $N_D(v_j)$ respectively. Consider node v_j. If it is decoding, $X_D(v_j) = X(v_1)$. If it is forwarding,

$$X_D(v_j) = \sqrt{\frac{P}{E\|Y_D(v_j)\|^2}}\ Y_D(v_j) = \sqrt{\frac{P}{P_D(v_j) + N_D(v_j)}}\ Y_D(v_j).$$

We now outline a method for finding the rate for all the nodes by proceeding in the order given by $\mathcal{S}$. Without loss of generality, assume that the nodes are already numbered according to a partial ordering. Therefore $\mathcal{S} = (v_1 = s, v_2, \ldots, v_V = d)$. Then, for v_2, we only have an edge coming in from s and hence

$$Y_D(v_2) = h_{1,2}X(v_1) + n_2.$$

Let our induction hypothesis be that we know $Y_D(v_j)$ for $j = 1, \ldots, i-1$. For $Y_D(v_i)$

we now have

$$\begin{aligned} & Y_D(v_i) \\ = \; & n_i + \sum_{v_j:(v_j,v_i)\in\mathcal{E}} h_{j,i} X_D(v_j) \qquad (5.4) \\ = \; & n_i + \sum_{v_j:(v_j,v_i)\in\mathcal{E}, v_j\in D\cup\{s\}} h_{j,i} X(v_1) + \sum_{v_j:(v_j,v_i)\in\mathcal{E}, v_j\notin D\cup\{s\}} h_{j,i} X_D(v_j) \\ = \; & n_i + \sum_{v_j:(v_j,v_i)\in\mathcal{E}, v_j\in D\cup\{s\}} h_{j,i} X(v_1) + \sum_{v_j:(v_j,v_i)\in\mathcal{E}, v_j\notin D\cup\{s\}} h_{j,i} \sqrt{\frac{P}{P_D(v_j)+N_D(v_j)}}\; Y_D(v_j). \end{aligned}$$

By our hypothesis, we know all the $Y_D(v_j)$ that occur in the last summation, Substituting for these, we get $Y_D(v_i)$. Careful observation indicates that this will be a linear combination of $X(v_1)$ and the noise terms $n_2, \ldots, n_i$.

In general, if this linear combination is given by

$$Y_D(v_i) = a_D X(v_1) + \sum_{j=2}^{i} a_{D,j}(v_i) n_j,$$

we have $P_D(v_i) = a_D^2 P$ and $N_D(v_i) = \sum_{j=2}^{i} a_{D,j}^2(v_i)\sigma_j^2$. Once these are known, the SNR is simply $\rho_D(v_i) = \frac{P_D(v_i)}{N_D(v_i)}$ and the rate can be calculated as $R_D(v_i) = \log(1 + \rho_D(v_i))$. Clearly, the complexity of this procedure is $O(V)$.

5.6.3 Finding the Rate in Erasure Wireless Networks

We first put this problem in a graph-theoretic setting. We are given a directed, acyclic graph where certain nodes act as sources. For us, the set $D \cup \{s\}$ is the set of source nodes. All the edges of the graph have certain probabilities of failing, i.e., of being absent. For us, these are the erasure probabilities of the channel. With this setup, for every node v in the network (excluding s, but including those in D) we need to find the probability that there exists at least one directed path from some source node to this node. This is the network reliability problem in one of its most general formulations [77, 76]. This is a well-studied problem and is known to be $\#P$-hard [76]. Although no polynomial-time algorithms to solve the problem are known, efficient algorithms

for special graphs are known. An overview of the network reliability problem can be found in [75]. In the rest of this section we propose two straightforward methods to compute the probabilities of connectivity that we are interested in. We will also mention some techniques that can reduce the computation involved in these methods.

Assume that we have a policy D. Consider a node v_i of the network. To find $R_D(v_i)$ we need to find $e_D(v_i)$. A bit is erased at node v_i if it is erased on all incoming links. With each edge (v_i, v_j) in the graph, associate a channel random variable $z(i,j)$. This takes the value 0 when a bit is erased and the value 1 when a bit is not erased. Thus, it is a Bernoulli random variable with probability $(1 - \epsilon_{i,j})$.

Consider all the directed paths from s to v_i. Let there be k_i paths. Denote the paths by $B_1, \ldots, B_{k_i}$. Let path B_j consist of l_j edges. We specify path B_j by writing in order the edges it traverses, i.e., with the sequence $((v_{j_1}, v_{j_2}), (v_{j_2}, v_{j_3}), \ldots, (v_{j_{l_j}}, v_{j_{l_j+1}}))$. We know that $s = v_{j_1}$ and $v_i = v_{j_{l_j+1}}$. Consider the set of vertices excluding v_i that are on path v_j, i.e., $\{v_{j_i} : i = 1, \ldots, l_j\}$. Some nodes in this set may belong to D, i.e., they are decoding nodes. In this case we know that they transmit the original codeword exactly. Let t be the largest index in this set such that v_{j_t} decodes. Therefore, v_i will not receive bit b along path B_j only if an erasure occurs on an edge that comes after v_{j_t} in the path. We associate with path B_j the product of the random variables that affect this, viz.,

$$Z_j = z(j_t, j_{t+1}) \cdot z(j_{t+1}, j_{t+2}) \cdot \cdots \cdot z(j_{l_j}, j_{l_j+1}).$$

This product is zero if one of the z random variables takes value zero, which, in turn, means that an erasure occurred on that edge.

Now, v_i sees an erasure only when none of the paths from s to itself manage to transmit the bit to it. Therefore, v_i sees an erasure when $Z_j = 0$ for *all* the paths

$B_j, j = 1, \ldots, k_i$. Therefore we have

$$\begin{aligned} R_D(v_i) &= 1 - e_D(v_i) \\ &= 1 - P(\bigcap_{j=1}^{k_i}(Z_j = 0)) \\ &= P(\bigcup_{j=1}^{k_i}(Z_j \neq 0)). \end{aligned}$$

One way to evaluate this is by checking all possible combinations of values that the z variables can take and finding the total probability of those combinations that satisfy $\cup_{j=1}^{k_i}(Z_j \neq 0)$. This procedure has complexity $O(2^E)$. One observation that can make this procedure more efficient is the following – if we know that setting a certain subset of the z variables to one is enough to make the event $\bigcup_{j=1}^{k_i}(Z_j \neq 0)$ happen, then for every superset of this subset, setting all the z variables in that superset to one is also enough to make the event $\bigcup_{j=1}^{k_i}(Z_j \neq 0)$ happen. With this, we may have to check out fewer than the 2^E possible combinations of values for the z variables and reduce the complexity.

Another way to evaluate this is by using the Inclusion Exclusion Principle [72]. This gives us

$$P(\bigcup_{j=1}^{k_i} Z_j \neq 0) = \sum_{r=1}^{k_i} \sum_{1 \leq j_1 < \cdots < j_r \leq k_i} (-1)^{r+1} P(Z_{j_1} \neq 0, \ldots, Z_{j_r} \neq 0).$$

Since we have k_i paths, the above expression has $2^{k_i} - 1$ terms. A general term of the form $P(Z_{j_1} \neq 0, \ldots, Z_{j_r} \neq 0)$ can be evaluated by first listing all the z variables that occur in at least one of the r terms. Say these are $z(i_1, j_1), \cdots, z(i_q, j_q)$. Now $P(Z_{j_1} \neq 0, \ldots, Z_{j_r} \neq 0)$ is given by the product $(1 - \epsilon_{i_1,j_1}) \times \cdots \times (1 - \epsilon_{i_q,j_q})$. This procedure has complexity $O(E2^k)$ where k is the $\max_i k_i$. In this procedure, the complexity of listing all the variables in a certain set of r terms can be reduced by storing the lists that one makes for sets of $(r-1)$ terms and simply adding on the z terms from the rth term to the appropriate list.

5.7 Algorithm to find Optimum Policy

In general, since we have $V-2$ relay nodes and each node has two options, viz., "forwarding" and "decoding and re-encoding," we have 2^{V-2} policies. To find the optimum policy we can analyze the rate for each of these policies and determine the one that gives us the best rate. This strategy of exhaustive search requires us to analyze 2^{V-2} policies.

Here, we propose a greedy algorithm that finds the optimum policy D which maximizes the rate. This algorithm requires us to analyze at most $V-2$ policies. In the next section we will give a proof of correctness for this algorithm.

(a) Set $D = \emptyset$.

(b) Compute $R_D(v_i)$ for all $v_i \in \mathcal{V}$. (Use techniques of Section 5.6.)
Find $R_D = \min_{v_i \in D \cup \{d\}} R_D(v_i)$.

(c) Find $M = \{v_i | v_i \notin \{s, d\} \cup D, R_D \leq R_D(v_i)\}$.

(d) If $M = \emptyset$, terminate. D is the optimal strategy.

(e) If $M \neq \emptyset$, find the largest $D' \subseteq M$ such that $\forall v \in D'$, $R_D(v) = \max_{v_i \in M} R_D(v_i)$.
Let $D = D \cup D'$.
Return to 2.

At each stage of the algorithm, we look for nodes that are seeing a rate as good as or better than the current rate of network operation. If there are no such nodes, the algorithm terminates. If there are such nodes, we choose the best from among them. Thus, in every iteration, the nodes we add are such that they do not put additional constraints on the rate of the network. Therefore, the rate of the network can only increase in successive iterations.

Note that since we assume a finite network, this algorithm is certain to terminate. Also, since D cannot have more than $(V-2)$ nodes, the algorithm cycles between steps 2 to 5 at most $(V-2)$ times. This is significantly faster than the strategy of exhaustive search that requires us to analyze 2^{V-2} policies.

The complexity of the algorithm depends on how fast the computation of $R_D(v_i)$ can be done. We have seen techniques for this computation in Section 5.6.

5.8 Analysis of the Algorithm

We first prove a lemma regarding the effect of decoding at a particular node on the rates supportable at other nodes.

Lemma 5.2. *When node v is added to the decoding set D, the only nodes v_i that may see a change in rate are $v_i > v$. This change can only be an increase in rate, i.e., $\forall v_i$ such that $v_i > v$ we have, $R_D(v_i) \leq R_{D\cup\{v\}}(v_i)$. Every other node v_j is unaffected, i.e., $R_D(v_j) = R_{D\cup\{v\}}(v_j)$.*

Proof. We give a proof for the Gaussian network. We omit the proof for erasure networks since it uses the same ideas.

Gaussian Network : Recall the computation of $\rho_D(v_i)$ described in Section 5.6.2. The computation for $Y_D(v_i)$ depends only on (some of) the $Y_D(v_j)$ where (v_j, v_i) is an edge. Therefore, inductively, it is clear that $Y_D(v_i)$ (and hence $\rho_D(v_i)$) depends only on the nodes v where $v < v_i$. Therefore, the only nodes that are affected when v changes its operation (from "forwarding" to "decoding and re-encoding") are $v_i > v$. The rest are unaffected.

Consider one of the $X_D(v_j)$ terms in (5.4). Note that each of these is of power P, of which some power is the signal power and the rest is the noise power. If v_j changes its operation from forwarding to decoding, $X_D(v_j) = X(v_1)$, i.e., the signal power increases to P and the noise power goes to 0. If v_j is forwarding, $X_D(v_j)$ is only a scaled version of $Y_D(v_j)$. Since it is always of power P, if the SNR at node v_j increases, the signal power in $X_D(v_j)$ increases while the noise power decreases. From (5.4) we see that in both these cases, there is an increase in the signal power of $Y_D(v_i)$ and a decrease in the noise power. This implies an increase in the SNR.

Therefore, when v is added to D, by induction, for all nodes $v_i > v$, the SNR, if affected, can only undergo an increase. Naturally, we have the same conclusion for the rate. □

This lemma tells us that adding nodes to the set of decoding nodes can only increase the rate to other nodes. While this sounds like a good thing, it also puts

a constraint on the rate as indicated by (5.2). It is this tradeoff that our algorithm seeks to resolve by finding the optimal set of decoding nodes.

5.8.1 Proof of Optimality

Theorem 5.3. *The algorithm of Section 5.7 gives us an optimal set of decoding nodes.*

Proof. Let S be an optimal set of decoding nodes. Let D be the set returned by the algorithm. We will prove that $R_D \geq R_S$. Then, since S is optimal, we will have $R_D = R_S$.

We prove $R_D \geq R_S$ in two steps. First we show that $R_{S\cup D} \geq R_S$. Then we show that $S \cup D - D = \emptyset$, i.e., $S \cup D = D$. This will complete the proof.

Step 1: In every iteration, the algorithm finds subsets D' and adds them to D. Denote by D_i the subset that is added to D in the i-th iteration. Assuming the algorithm goes through m iterations, we have $D = D_1 \cup \cdots \cup D_m$ where the union is over disjoint sets. In the algorithm, when D_i is added to D, all the nodes in it are decoding at the same rate which is $R_{D_1\cup\cdots\cup D_{i-1}}(v)$ for $v \in D_i$. We will call this rate $R_{\text{algo},i}$. Consider the smallest i such that $D_i \nsubseteq S$, i.e., D_i is not already entirely in S.

Claim: Adding D_i to S does not decrease the rate, i.e., $R_{S\cup D_i} \geq R_S$.

Proof. Because of the acyclic assumption on the graph, we will have some nodes $v \in S$ such that $\forall u(\neq v) \in S$ we either have $v < u$ or v and u are incomparable. Let L be the set of all such nodes v. Note that by Lemma 5.2, node v supports a rate $R_S(v) = R_\emptyset(v)$. By (5.3), for every $v \in L$ we have the necessary condition

$$R_S \leq R_S(v) = R_\emptyset(v). \tag{5.5}$$

Also note that $D_1, \ldots, D_{i-1}$ are all in S and by the definition of L and Lemma 5.2 we have

$$R_\emptyset(v) = R_{D_1\cup\cdots\cup D_{i-1}}(v). \tag{5.6}$$

We now consider two cases.

- If for some $w \in L$, we also have $w \in D_i$, then from (5.5) and (5.6) we have $R_S \leq R_S(w) = R_\emptyset(w) = R_{D_1 \cup \cdots \cup D_{i-1}}(w) = R_{\text{algo},i}$.

- On the other hand, if none of the nodes in L is in D_i, pick any node $v \in L$. We have $v \notin D_i$. We now consider two subcases.

  - Let $v \notin D_1, \ldots, D_{i-1}$. We note from steps 3 and 5 of the algorithm that it picks out from the set of nodes not in D, all nodes with the best rate. Since v does not get picked, we have $R_{\text{algo},i} > R_{D_1 \cup \cdots \cup D_{i-1}}(v)$. This along with (5.5) and (5.6) gives us $R_S \leq R_{D_1 \cup \cdots \cup D_{i-1}}(v) < R_{\text{algo},i}$.

  - The other possibility is that $v \in D_1 \cup \cdots \cup D_{i-1}$. Since the D_is are disjoint, there is a unique j such that $v \in D_j$. Since $v \in L$, by Lemma 5.2, $R_{\text{algo},j} = R_{D_1 \cup \cdots \cup D_{j-1}}(v)$. With the same argument as that for (5.6), we have $R_\emptyset(v) = R_{D_1 \cup \cdots \cup D_{j-1}}(v)$. But since the algorithm never decreases rate from one iteration to the next, we have $R_{\text{algo},i} \geq R_{\text{algo},j}$. Putting these together we get $R_{\text{algo},i} \geq R_{\text{algo},j} = R_{D_1 \cup \cdots \cup D_{j-1}}(v) = R_\emptyset(v)$. With (5.5) this gives us $R_S \leq R_S(v) = R_\emptyset(v) \leq R_{\text{algo},i}$.

Therefore, in every case, we have shown that $R_S \leq R_{\text{algo},i}$. This implies that adding the rest of the nodes from D_i to S will not put additional constraints on R_S and hence cannot decrease the rate. Therefore we have $R_{S \cup D_i} \geq R_S$. □

Since S is optimal, this proves that $S \cup D_i$ also achieves optimal rate. We can now call this set S and for the next value of i such that $D_i \nsubseteq S$, we can prove that $S \cup D_i$ has optimal rate. Continuing like this we have that $S \cup D$ is optimal, or, in other words, $R_{S \cup D} \geq R_S$.

Step 2: Next we wish to show that $S \subseteq D$, i.e., $S \cup D - D = \emptyset$. Let us assume the contrary. Let $T = S \cup D - D$. Therefore, $T \cap D = \emptyset$ but $T \subseteq S$. Thus, $D \cup S = D \cup T$ where D and T are disjoint. Consider $v \in T$ such that $\forall u (\neq v) \in T$, we either have $v < u$ or v and u are incomparable. We have $R_{D \cup T}(v) = R_{D \cup S}(v)$. By Lemma 5.2, $R_{D \cup T}(v) = R_D(v)$. Also, the constraint of (5.2) tells us that $R_{D \cup S} \leq R_{D \cup S}(v)$. Finally, note that since the algorithm terminates without adding v to D, we have

$R_D > R_D(v)$. Putting these inequalities together we have $R_D > R_D(v) = R_{D\cup T}(v) = R_{D\cup S}(v) \geq R_{D\cup S}$. But this contradicts the fact that $S \cup D$ is optimal. Thus we have $S \subseteq D$, i.e., $S \cup D = D$.

From steps 1 and 2 we have $R_D \geq R_S$. But since S was an optimal policy, D is also an optimal policy. This proves that the algorithm does indeed return an optimal set of decoding nodes.

The only case in which this proof does not go through is when the algorithm returns $D = \emptyset$ and $S \neq \emptyset$. In this case, consider node $v \in L \subseteq S$, where L is as defined earlier. Since the algorithm does not pick up v, we have $R_\emptyset > R_\emptyset(v)$. But $R_S \leq R_S(v) = R_\emptyset(v)$ from (5.5). Thus, $R_S < R_\emptyset$. But this contradicts the optimality of S. Therefore, if there exists an optimal, non-empty S, the algorithm cannot return an empty D. □

Corollary 5.4. *The algorithm of Section 5.7 returns the largest optimal policy D.*

Proof. In the proof above, we have shown that for any optimal policy S, we have $S \subseteq D$. This implies that D is the largest optimal policy. □

5.9 Examples

In this section we present some examples of networks and show how the algorithm runs on them.

5.9.1 Multistage Erasure Relay Networks

In Figure 5.4(a) we have depicted a multistage relay network. In this we have a single source and destination and k layers of relay nodes. The ith layer consists of l_i nodes. Between the ith and the $(i+1)$th layer we have a complete bipartite graph where all the edges are directed from the ith layer to the $(i+1)$th. We assume that each of these edges has erasure probability ϵ_i. The source is connected to all the nodes in the first layer by erasure channels with erasure probability ϵ_0 and all the nodes in the kth layer are connected to the destination by erasure channels with erasure probability ϵ_k. We will also call d the $(k+1)$th layer and $l_{k+1} = 1$.

Because of the structure of this network, finding the rate under a particular policy is easier than indicated in Section 5.6.3. Denote by $Q_{i,j}$ the probability that in layer i there are j nodes that do not see an erasure. This defines $Q_{i,j}$ for $i = 1, 2, \ldots, (k+1)$ and $j = 0, 1, \ldots, l_i$. With this, for $i = 1$ we obtain

$$Q_{1,k} = \binom{l_1}{k} \epsilon_0^{l_1-k} (1-\epsilon_0)^k. \tag{5.7}$$

For $i > 1$, we can show the recursion below.

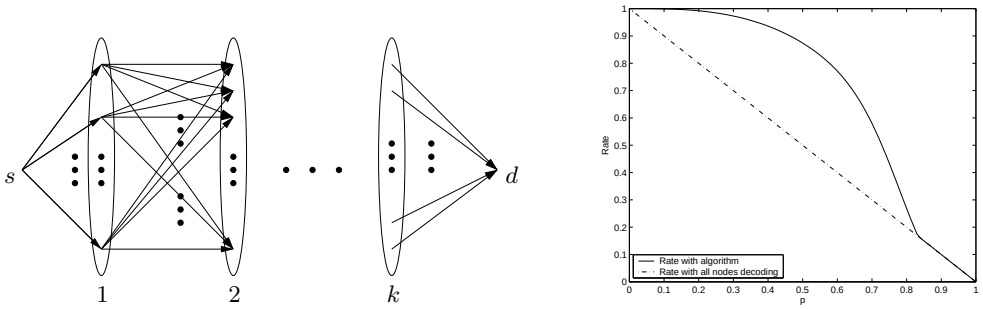
$$Q_{i,k} = \binom{l_i}{k} \sum_{t=0}^{l_{i-1}} \epsilon_{i-1}^{t(l_i-k)} (1-\epsilon_{i-1}^t)^k Q_{i-1,t}. \tag{5.8}$$

Denote by e_i the probability that the at least one node in the ith layer does not see an erasure. We can show that

$$e_i = \sum_{k=0}^{l_i} Q_{i,k} \left(1 - \frac{k}{l_i}\right).$$

Note that, by symmetry, whenever a node decides to decode, all the nodes in that layer decode. When layer i decides to decode, we set $Q_{i,l_i} = 1$ and $Q_{i,j} = 0$ for $j \neq l_i$ and continue with the recursion of (5.8) for the other layers. This also extends to the case when more than one layer decodes.

Now, our algorithm proceeds as before, but operates on layers rather than nodes and the effective erasure probability at layer i is e_i. As an explicit example, consider a multistage relay network with four layers between the source and destination. Let $l_1 = 3, l_2 = 6, l_3 = 4, l_4 = 5$ and $\epsilon_0 = p, \epsilon_1 = p^2, \epsilon_2 = p, \epsilon_3 = p^3, \epsilon_4 = p$ where p is any number in the interval $[0, 1]$. For a fixed value of p, we can find the optimum policy for the network and this will give us the optimal rate. Figure 5.4(b) shows this optimal rate for the parameter p going from 0 to 1 (solid curve). This is not a smooth curve. The point where the right and left derivatives do not match is where either the optimum policy or the rate-determining layer changes. The rate with all nodes decoding has also been plotted (dashed curve). This rate is $1 - p$ and we see



(a) Model of a multistage relay network

(b) Maximum rate as given by the algorithm

Figure 5.4: Multistage relay network. Rate achieved by the optimum forward/decode scheme is greater than the rate achieved when all nodes decode.

that the algorithm gives us dramatically higher rates.

5.9.2 Multistage Gaussian Relay Networks

We consider a multistage network similar to the one of the previous section, but in which the the links represent Gaussian channels with fading coefficients h_i and with additive noise σ_i^2 at layer i. The indexing is identical to that in the erasure network.

Because of the structure of the network, it is easy to compute SNRs. Let $\rho(i)$ denote the SNR at layer i. Then, in the situation where all the nodes are forwarding, the following recursion gives us the SNR. We initialize the recursion as follows.

$$a(1) = h_0^2 P \quad b(1) = \sigma_1^2 \quad \rho(1) = \frac{a(1)}{b(1)}.$$

For the rest of the layers, i.e., $i \geq 2$ we have

$$\begin{aligned} a(i) &= a(i-1)\frac{h_i^2 l_i^2}{1+\frac{1}{\rho(i-1)}}; \\ b(i) &= b(i-1)\frac{h_i^2 l_i}{1+\frac{1}{\rho(i-1)}} + \sigma_i^2 i; \\ \rho(i) &= \frac{a(i)}{b(i)}. \end{aligned}$$

As with the erasure relay network, whenever a node decides to decode, all the nodes in that layer decode. If some layers decide to decode, a simple modification of the above recursion gives us the new rates. If i is the smallest number such that the ith layer decodes, then, clearly, the above recursion gives us rates for layers l_1 to l_i. For l_{i+1}, we set $a(i+1) = h_i^2 l_i^2 P$ and $b(i+1) = \sigma_{i+1}^2$. We have $\rho(i+1) = a(i+1)/b(i+1)$ as before and we can continue with the recursion above for layers $(i+2)$ etc. We repeat this modification for each layer that decodes.

Once the SNR at a layer is known, the rate is given by $\log(1+\rho)$ as usual. With this procedure for calculating rates, we use the algorithm of Section 5.7. It now operates on layers rather than nodes.

As an explicit example, consider a multistage relay network with three layers between the source and destination. Each node is restricted to using power $P = 1$. Let $l_1 = 2, l_2 = 5, l_3 = 3$ and $h_0 = 0.7, h_1 = 10, h_2 = 0.1, h_3 = 1$. We will have $\sigma_1^2 = m^2, \sigma_2^2 = m, \sigma_3^2 = m^3, \sigma_4^2 = m^2$ where m can be any positive real number. For a fixed value of m, we can find the optimum policy for the network and this will give us the optimal rate. Figure 5.5 shows this optimal rate for the parameter m going from 0.5 to 1.5 (solid curve). As with the multistage erasure network, the curve is not smooth at points where the optimum policy or the rate-determining layer changes. We also see the advantage compared to the case when all nodes decode (dashed curve).

5.9.3 Erasure Network with Four Relay Nodes

Consider the relay network of Figure 5.6(a). All the links have the same erasure probability p, where p is any number between 0 and 1. For this range of p, the algorithm has been used to find the optimum rates and policies. The rate is plotted in Figure 5.6(b) (solid curve). Throughout, the optimal policy is $D = \{v_2, v_3, v_5\}$. The rate with all nodes decoding is $1 - p$ and is also plotted (dashed curve). As expected, the algorithm outperforms the all-decoding scheme.

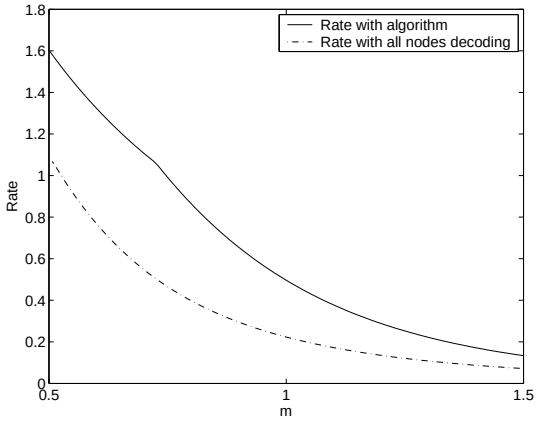
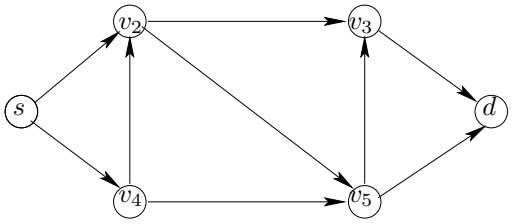
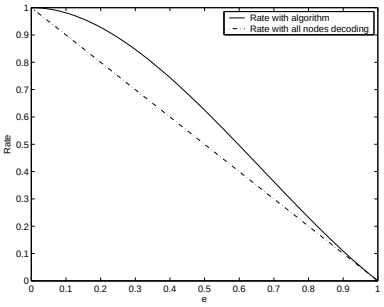


Figure 5.5: Rate for the multistage Gaussian relay network. Rate achieved by the optimum forward/decode scheme is greater than the rate achieved when all nodes decode.



(a) Erasure network with four relay nodes.

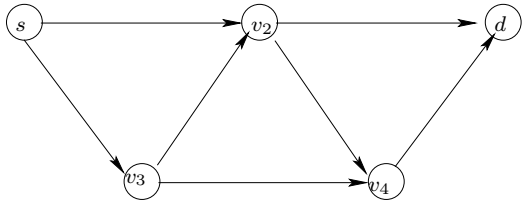


(b) Maximum rate as given by the algorithm

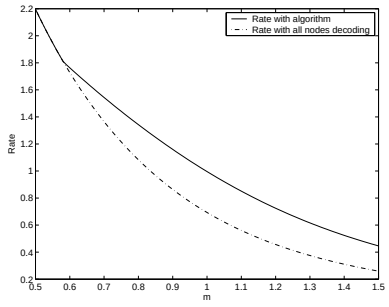
Figure 5.6: Erasure network with four relay nodes. Rate achieved by the optimum forward/decode scheme is greater than the rate achieved when all nodes decode.

5.9.4 Gaussian Network with Three Relay Nodes

In Figure 5.7(a) we see a Gaussian network with three relay nodes. We assume that each node is restricted to use power $P = 1$. Let the additive noise variances be $\sigma_2^2 = m, \sigma_3^2 = m^3, \sigma_4^2 = m^2, \sigma_5^2 = m^1$ where m can be an arbitrarily chosen real number. In Figure 5.7(b) we see the rate returned by the algorithm for the optimal policy for $m \in [0.5, 1.5]$ (solid curve). The rate with all nodes decoding is also plotted (dashed curve). In the region $m \in [0.5, 0.58]$ we see that the optimal policy is in fact that of decoding at all nodes and the two curves match. After that, the optimal



(a) Gaussian network with three relay nodes.



(b) Maximum rate as given by the algorithm

Figure 5.7: Gaussian network with three relay nodes. Rate achieved by the optimum forward/decode scheme is greater than the rate achieved when all nodes decode.

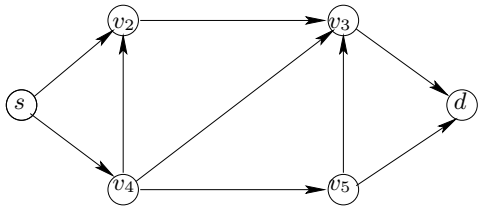
policy changes and hence we see that the optimal rate curve is not smooth.

5.9.5 Gaussian Network with Four Relay Nodes

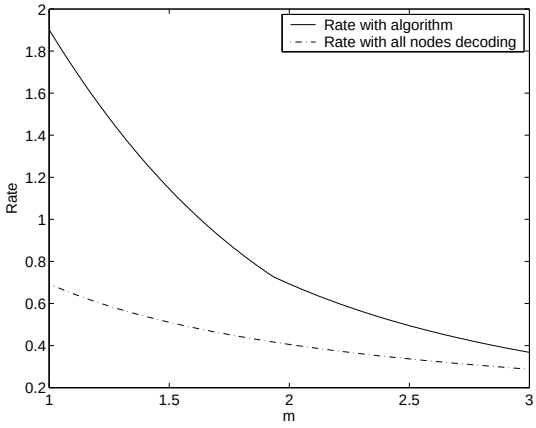
In Figure 5.8(a) we see a Gaussian network with four relay nodes. Each node, including the source, is restricted to using power $P = 1$. The attenuation factors associated with the edges are $h_{1,2} = 1, h_{1,4} = 2, h_{4,2} = 3, h_{2,3} = 4, h_{4,3} = 5, h_{4,5} = 1, h_{3,6} = 3, h_{5,3} = 2, h_{5,6} = 4$. The additive noise variances associated with the nodes are $\sigma_2^2 = m, \sigma_3^2 = m^3, \sigma_4^2 = m^2, \sigma_5^2 = m, \sigma_6^2 = m^3$ where m can be any positive real number. In Figure 5.8(b) we see the rate returned by the algorithm for the optimal policy for $m \in [1, 3]$ (solid curve). The rate with all nodes decoding is also plotted (dashed curve). We see that the forward/decode scheme gives us significant improvements in the rate.

5.10 A Distributed Algorithm for the Optimal Policy

The algorithm as proposed in Section 5.7 requires that the network parameters (viz., noise variances or erasure probabilities) be known before the network operation begins



(a) Gaussian network with four relay nodes.



(b) Rate for the Gaussian relay network with four relay nodes as given by the algorithm (solid curve) is much higher than that with all nodes decoding (dashed curve)

Figure 5.8: Gaussian network with four relay nodes. Rate achieved by the optimum forward/decode scheme is greater than the rate achieved when all nodes decode.

so that the optimum policy is known beforehand. With the algorithm in its current form the nodes cannot determine for themselves if they should decode or forward. In this section we propose a scheme that can permit nodes to determine their own operation.

The algorithm works iteratively to converge to a rate. In each iteration, the rate of operation of the network is incremented or decremented depending on whether the previous transmission was successful or not. In every iteration, all the nodes get to decide their operation for themselves.

Let R^* be the maximum rate of the network. This is not known beforehand. We assume that parameters R, δ and N are known to all the nodes beforehand. The blocklength n is also predetermined and known to all the nodes. In addition, we require that the nodes have a common source of randomness so that they can generate the *same* random codebook individually. With this, consider the following algorithm.

(a) All nodes generate the (same) codebook for rate R. They all set $k = 0$.

(b) s transmits a randomly chosen codeword $X(v_1)$.

(c) Every relay node v_i attempts to decode the received message $Y(v_i)$.
If it can decode without error, it transmits the decoded codeword. [2]
Else, it forwards the received message (with appropriate scaling, for the Gaussian network).

(d) The destination attempts to decode the received message.
If it decodes without error, it sends back bit 1 to all the other nodes to indicate successful decoding.
Else, it sends back bit 0 to all other nodes.

(e) All nodes increment k. $k = k + 1$.
If transmitted bit was zero, all nodes set $R = R - \delta/2^k$.
If transmitted bit was one, all nodes set $R = R + \delta/2^k$.

(f) While $k \leq N$, go to step 1.

Theorem 5.5. *If the maximum rate of the network, viz. R^* is in the range $[R - \delta, R + \delta]$, the algorithm above converges to it with an accuracy of $\frac{\delta}{2^N}$.*

Proof. The source starts by transmitting at rate R. Each relay node receives messages on all incoming links and decodes the message if it can. If it cannot, it simply forwards what it has received. With this procedure, nodes decide their own operation. (The order in which they decide this is a partial order in the sense defined in Section 5.6.1.) After the destination receives all its incoming messages, it tries to decode. If $R > R^*$, the destination will definitely not be able to decode. If $R \leq R^*$, we claim that the destination will be able to decode. This is because when a node decodes, it only improves the rates for other nodes. Also, note that an arbitrary node v decides whether to decode or not only after all the nodes before it in the partial order have already determined if the rate they can support is greater or smaller than R. Since, by Lemma 5.2, these are the only nodes that affect the rate for v and they decode whenever they can, node v always gets to see the best situation it can as far as rate R is concerned. This is true for the destination also.

[2] One method of error detection is for a node to perform typical set decoding, and assume an error if it finds more than one codeword that is jointly typical with the received message. Other methods of error detection are the introduction of cyclic redundancy checks (CRCs) or an ARQ protocol e.g. [69].

Therefore, depending on whether the destination can decode or not, we can say if R^* is greater or smaller than R. If this bit of information is transmitted back to the source and other nodes, they can accordingly decide whether to increase or decrease the rate for the next transmission. Thus we have a decision tree of rates such that the ability or inability of the decoder tells us which path to traverse in that tree. With this we can finally converge on a rate sufficiently close to the actual rate R. □

This algorithm provides a very natural mode of network operation that obviates the need for a central agent to know the entire network and decide the optimum policy. Although some communication from the destination to the source and other nodes is required, this is minimal and should be easily possible in a practical network setting.

We mention that the algorithm we present can be made more sophisticated such that it works for all values of R^*, rather than just those in the interval $[R-\delta, R+\delta]$. We omit the details in the interests of brevity.

5.11 Upper Bounds on the Maximum Rate

The algorithms of Section 5.7 as well as Section 5.10 converge to the maximum rate possible with the decode/forward scheme, but we have no way of simply looking at the network and saying what this maximum rate will be. In this section, we present upper bounds on the rate achievable with the limited operations that we use in this chapter.

5.11.1 Definitions

An $s-d$ *cut* is defined as a partition of the vertex set $\mathcal{V}$ into two subsets $\mathcal{V}_s$ and $\mathcal{V}_d = \mathcal{V} - \mathcal{V}_s$ such that $s \in \mathcal{V}_s$ and $d \in \mathcal{V}_d$. Clearly, an $s-d$ cut is determined simply by $\mathcal{V}_s$. For the $s-d$ cut given by $\mathcal{V}_s$, let the *cutset* $\mathcal{E}(\mathcal{V}_s)$ be the set of edges defined below

$$\mathcal{E}(\mathcal{V}_s) = \{(v_i, v_j) | (v_i, v_j) \in \mathcal{E}, v_i \in \mathcal{V}_s, v_j \in \mathcal{V}_d\}$$

Finally, we define $X(\mathcal{V}_s)$ and $Y(\mathcal{V}_s)$ as below.

$$X(\mathcal{V}_s) = \{v_i | (v_i, v_j) \in \mathcal{E}(\mathcal{V}_s)\} \qquad Y(\mathcal{V}_s) = \{v_j | (v_i, v_j) \in \mathcal{E}(\mathcal{V}_s)\}$$

Thus $X(\mathcal{V}_s)$ and $Y(\mathcal{V}_s)$ denote the nodes transmitting and receiving messages across the cut, respectively.

5.11.2 Upper Bound for Gaussian Networks

For Gaussian networks, it is evident that making the additive noise zero at certain nodes can only increase the maximum rate available at d. In particular let us make the additive noise zero at all nodes except $Y(\mathcal{V}_s)$. Therefore, the received messages (and the transmitted messages) at all nodes in $\mathcal{V}_s$ are exactly the same as that transmitted by the source. Now, if we permit the nodes in $Y(\mathcal{V}_s)$ to decode cooperatively, the rate at which they can decode will give us an upper bound on the rate that the destination can get.

Note that the SNR at node $v_j \in Y(\mathcal{V}_s)$ is

$$\frac{P}{\sigma_j^2} \left(\sum_{v_i:(v_i,v_j)\in\mathcal{E}(\mathcal{V}_s)} h_{i,j} \right)^2 .$$

Since our codebook and noise are Gaussian distributed, the optimum scheme for decoding cooperatively is taking a suitable linear combination of received messages and then decoding that. For optimal decoding, we find the linear combination that gives us the best SNR. It is easy to show that the best SNR possible is the sum of the SNRs seen by each node in $Y(\mathcal{V}_s)$.

Therefore, an upper bound on the rate is

$$R \leq \log \left(1 + \sum_{v_j \in Y(\mathcal{V}_s)} \frac{P}{\sigma_j^2} \left(\sum_{v_i:(v_i,v_j)\in\mathcal{E}(\mathcal{V}_s)} h_{i,j} \right)^2 \right)$$

for every cut $\mathcal{V}_s$.

5.11.3 Upper Bound for Erasure Networks

As in the above section, we can obtain an upper bound on the rate for erasure networks by making certain links perfect, or free of erasures. Therefore we can obtain an upper bound on the rate by making all edges other than those in $\mathcal{E}(\mathcal{V}_s)$ perfect. With this all the received (and transmitted) messages in $\mathcal{V}_s$ are exactly the same as the codeword transmitted by the source. Now, it is clear that the rate at which the nodes in $Y(\mathcal{V}_s)$ can decode co-operatively is an upper bound on the rate available at the destination.

Clearly, the effective erasure probability seen by the set of nodes $Y(\mathcal{V}_s)$ is $\prod_{(v_i,v_j)\in\mathcal{E}(\mathcal{V}_s)} \epsilon_{i,j}$ This gives us an upper bound on the rate. We have

$$R \leq 1 - \prod_{(v_i,v_j)\in\mathcal{E}(\mathcal{V}_s)} \epsilon_{i,j}$$

for every cut $\mathcal{V}_s$.

Note that in [63], a different min-cut upper bound is proposed and is shown to be achievable. This gives the capacity of the network under the assumption that the destination has perfect side-information regarding erasure locations from across the network. This is very different from the setup of this chapter.

5.12 Conclusions

As the previous discussion demostrates, making each link in a wireless network error-free is sub-optimal. Thus a multihop approach, in which every relay node decodes the received message, is not necessarily a good approach for wireless networks. Restricting attention to nodes that perform either decoding or forwarding, the proposed algorithms include a greedy centralized algorithm for finding an optimal code and a distributed algorithm that iteratively converges to an optimal solution without the benefit of a central decision-making agent.

The algorithm of Section 5.7 finds the maximal rate and optimal policy for any Gaussian or erasure wireless network. The results of Section 5.11 roughly bound the optimal rates. However, we still do not know what policies are optimal for each

erasure probability or SNR. The examples of Section 5.3 suggest that decoding is better when the links are poor (high erasure probabilities or low SNR). It would be interesting to know if this pattern holds for general networks and to find thresholds below which a certain operation is always preferred.

Corollary 5.4 proves that the algorithm returns the largest decoding set. Since decoding is the more costly of the two operations considered here, it would be useful to develop an algorithm that achieves the maximal rate using the smallest decoding set.

Both decoding and forwarding are special cases of network coding. We can also imagine a larger set of operations and consider optimal code design for more general code types. (In the most general case, all functions are allowed at a given node. This puts the problem in an information-theoretic setting and a general solution for erasure networks is proposed in the previous chapter, and [63], where network coding techniques are used to obtain the precise capacity region for several multicast settings in erasure networks, assuming certain side-information. Naturally, this capacity region is an upper bound for the rates we have obtained in the absence of this side-information. Finding practical schemes that reach this capacity is an interesting avenue for future work.

Chapter 6

Statistical Pruning for Near-Maximum Likelihood Decoding

In this chapter, we switch gears and present a problem in point-to-point communication, involving multiple antenna systems. In many such systems, maximum-likelihood (ML) decoding reduces to finding the closest (skewed) lattice point in N-dimensions to a given point $x \in \mathbb{C}^N$. This problem is known to be NP-complete in its full generality. The expected complexity of the sphere decoder, a particular algorithm that solves the ML problem exactly, has recently been computed, where it is shown that over a wide range of rates, SNRs and dimensions N, the expected computation involves no more than N^3 computations. In this chapter, we propose an algorithm that, for large N, offers substantial computational savings over the sphere decoder, while maintaining performance arbitrarily close to ML decoding. We statistically prune the search space to a subset that contains the optimal solution with high probability, thereby reducing the complexity of the search. We derive Bounds on the error performance of the new method and give both an upper bound and an approximate analysis of its complexity. The asymptotic behavior of the upper bound is also analyzed. Simulation results show that, the algorithm compares favorably in terms of computational complexity with the original sphere decoder without sacrificing much in terms of performance.

6.1 Introduction

Multiple antenna communication systems are capable of achieving high data rates. However, reliable decoding in these systems requires very high complexity. For a wide class of space-time transmission schemes (see e.g., [78, 79, 80]) ML decoding requires us to solve an integer least-squares problem. This is the problem of finding the closest (skewed) lattice point in N-dimensions to a given point $x \in \mathbb{C}^N$, which is known in general to be NP-hard. Most existing communications systems employ approximations or heuristics and typically require $O(N^3)$ operations (since underlying all of the methods is the calculation of a pseudo-inverse). Zero forcing cancellation, nulling and canceling and nulling and canceling with optimal ordering [78, 79, 81] are common techniques. The bit error rate (BER) performance of these algorithms is vastly inferior to that of the exact methods.

Exact methods require search over a space growing at an exponential rate. More sophisticated exact methods such as Kannan's algorithm [82], the KZ algorithm [83] and the sphere decoding algorithm of [84] attempt to reduce the search space. The branch and bound algorithm, popularly used to solve integer (usually linear) programming problems, can also be used [85]. However, branch and bound imposes additional constraints on the optimizing variables to reduce the size of the problem and also requires one to estimate upper and lower bounds for the objective function to prune the search tree. An improved sphere decoder based on the branch and bound method appears in [86].

In the sphere decoding algorithm we first determine all lattice points lying in a hypersphere centered at x and then determine the point closest to x. The complexity of the algorithm is therefore a function of the amount of work that is required to determine all lattice points inside a given hypersphere. (For some alternatives to sphere decoding see [83, 87, 88]). The sphere decoding algorithm requires exponential complexity in both worst-case and average analyses (see e.g., [89]). Since the noise vector and the lattice-generating-matrix are random, we can view the computational complexity as a random variable [90]. Analyzing the expected complexity of sphere

decoding, as well as its second-order moment, [90] shows that, over a wide range of rates, dimensions and SNRs, the algorithm uses no more than N^3 multiplications. While this result is very interesting, the expected number of operations still becomes prohibitively large for large enough N and low SNRs. This fact is formalized in [91] which shows that for any SNR the sphere decoder has exponential expected complexity.

In spite of this, the sphere decoder has attracted great interest and it has been proposed as the decoder for several space-time coded systems. In addition, several modifications to the sphere decoder have been suggested in the last few years that attempt to reduce the computation involved [69, 93, 92, 94, 95, 96]. Implementations of the sphere decoder in a complex setting rather than a real one are suggested in [90] and [99]. Some of the suggested modifications solve the ML decoding problem exactly ([69, 93, 94]) and others sacrifice some performance in order to reduce complexity ([95, 96]).

The efficiency of the sphere decoder demonstrates the power of the probabilistic viewpoint and we will continue to use it in the problem at hand. The main point is to understand the role of the randomness underlying the problem and leverage it suitably. We propose a modification to the sphere decoding algorithm that uses statistical pruning to reduce the exponentially large search space to one that is much smaller yet contains the optimal solution with high probability. This causes a significant reduction in complexity, at the price of a slight increase in the bit error rate (BER). We bound this loss of performance and describe methods for controlling it. We analyze the complexity in three ways. The first analysis is for asymptotically large systems and is of theoretical interest. The other two are valid for any value of N and can be used to design and understand practical systems.

The remainder of the chapter is organized as follows. In Section 6.2 we introduce the integer least-squares problem and demonstrate that it arises in the ML decoding of multiple antenna systems. In Section 6.3 the basic sphere decoding algorithm is explained and in Section 6.4 the notion of complexity is outlined. In Section 6.5, we introduce the statistics of the problem and propose a new algorithm, called

the Increasing Radii Algorithm, that exploits these statistics. (This algorithm was first presented in [96]). In Section 6.6 we bound the performance of this algorithm with respect to the optimal, or ML, performance and in Section 6.7 we analyze the complexity of the proposed algorithm. We then present simulations in Section 6.8. Ideas for future work and conclusions appear in Section 6.9.

6.2 Integer Least-Squares Problem

The integer least-squares problem is the following minimization problem:

$$\min_{s \in \mathbb{Z}^{M \times 1}} \|x - Hs\|^2$$

where $x \in \mathbb{C}^{N \times 1}$ and $H \in \mathbb{C}^{N \times M}$ are known and $\mathbb{Z}^{M \times 1}$ is the M-dimensional integer lattice. Often the search space is a finite subset of the integer lattice, say $\mathcal{A}$, in which case the minimization is done over $s \in \mathcal{A}$ rather than $s \in \mathbb{Z}^{M \times 1}$. This problem arises in several situations in communications, cryptography, etc. For a general H, it is known to be NP-hard in the worst-case sense [100] as well as the average sense [89, 101]. We now describe this problem in the context of ML decoding in a multiple antenna system.

6.2.1 System Model

We assume a discrete-time block-fading multiple antenna channel model with M transmit and N receive antennas, where the channel is known to the receiver. This is a reasonable assumption for communication systems where the signalling rate is much higher than the rate at which the propagation environment changes, so that the channel may be learned (perhaps by transmitting known training sequences) by the receiver. If $\mathcal{S}$ is the finite signal constellation, then during any channel use, the transmitted signal $\tilde{s} \in \mathcal{S}^{M \times 1}$ and the received signal $x \in \mathbb{C}^{N \times 1}$ are related by

$$x = H\tilde{s} + v \tag{6.1}$$

where $H \in \mathbb{C}^{N\times M}$ is the known channel matrix with independent, identically distributed (i.i.d.) complex Gaussian entries of variance σ_h^2, i.e., $\mathbb{C}\mathcal{N}(0,\sigma_h^2)$. We assume $N \geq M$ [1]. $v \in \mathbb{C}^{N\times 1}$ is the unknown additive noise vector, comprised of i.i.d. complex Gaussian entries of variance σ_v^2, i.e., $\mathbb{C}\mathcal{N}(0,\sigma_v^2)$. Without loss of generality, we assume $\sigma_v^2 = 1$. Thus, H and v are the only sources of randomness when $\tilde{s}$ is a particular transmitted point. With this setup we have $\sigma_h = \frac{\sigma_v}{\sigma_s}\sqrt{\frac{\rho}{M}}$ where ρ is the expected signal-to-noise ratio (SNR) and σ_s^2 is the average power of the signal constellation $\mathcal{S}$. Under the aforementioned assumptions the ML criterion requires us to find $s \in \mathcal{S}^{M\times 1}$ that minimizes $\|x - Hs\|^2$. This is equivalent to the integer least-squares problem mentioned in Section 6.2 where the search space, $\mathcal{A}$, viz. $\mathcal{S}^{M\times 1}$, is finite but has cardinality exponential in M.

This is different from the general integer least-squares problem in that H and v are random and hence the complexity of solving this problem is also a random variable. Therefore it is the various moments of the complexity that are of interest to us – we focus on the expected complexity in our work.

Also, the underlying probability distributions tell us how to prune the search space in order to reduce the complexity of the general integer least-squares problem while maintaining performance close to optimal.

In this chapter we only consider L^2-QAM constellations with even L, i.e.,

$$\mathcal{S} = \left\{ a + jb \mid a, b \in \left\{ -\frac{L-1}{2}, -\frac{L-3}{2}, \cdots, \frac{L-3}{2}, \frac{L-1}{2} \right\} \right\}. \tag{6.2}$$

It is then easy to show that $\sigma_s^2 = \frac{L^2-1}{6}$. This gives us $\sigma_h = \sqrt{\frac{6}{L^2-1}\frac{\rho}{M}}$.

Finally we note that the above description fits a system in which transmissions are uncoded. In the ML decoding of systems involving space-time codes etc., we also run into the integer least-squares problem [78, 79, 80]. In this situation, the operational meanings of M, N and H may be different since they now involve the coding scheme as well as the physical antennas. For instance, M and N would

[1] The case $N < M$ can also be dealt with using the approach of this chapter. However, since it inevitably requires an exhaustive search over a lattice of dimension $M - N$, we shall not consider it here.

typically be much larger than the actual number of transmit and receive antennas and H would have entries that are functions of the coding scheme and the channel values. (These would not necessarily be i.i.d. entries.) The algorithms mentioned in this chapter would work for these systems also, however the analysis of the performance and computational complexity would be different and would vary from system to system. The analysis of the i.i.d. case is complicated as is and would become even more intractable in the correlated case. Therefore we restrict the analysis to H matrices with i.i.d. entries. We deal with non-i.i.d. matrices through simulations where we run the proposed decoder on space-time coded systems that lead to an equivalent channel with correlation.

6.3 Sphere Decoder

In this section we introduce the sphere decoder and also introduce the notation that will be used in the rest of the chapter. In sphere decoding we search only over lattice points that lie in a hypersphere of radius r around x, thus reducing the search space and the computation. Therefore we first need to find all $s \in \mathcal{S}^{M\times 1}$ that lie within this hypersphere of radius r. This is equivalent to solving

$$r^2 \geq \|x - Hs\|^2. \tag{6.3}$$

To this end, consider the QR decomposition of the channel matrix $H = Q \begin{bmatrix} R \\ 0_{(N-M)\times M} \end{bmatrix}$, where R is an $M \times M$ upper triangular matrix with non-negative diagonal entries and Q is an $N \times N$ unitary matrix. Such a decomposition is unique. Partition Q as $\begin{bmatrix} Q_1 & Q_2 \end{bmatrix}$ where Q_1 is $N \times M$ and Q_2 is $N \times (N-M)$. Since Q is unitary, so is Q^*. We know that premultiplying by a unitary matrix does not change the squared-norm of a vector. Therefore (6.3) becomes:

$$r^2 \geq \|x - Hs\|^2 = \left\| x - Q \begin{bmatrix} R \\ 0 \end{bmatrix} s \right\|^2 = \left\| \begin{bmatrix} Q_1^* \\ Q_2^* \end{bmatrix} x - \begin{bmatrix} R \\ 0 \end{bmatrix} s \right\|^2 \tag{6.4}$$

Define

$$z = \begin{bmatrix} Q_1^* \\ Q_2^* \end{bmatrix} x - \begin{bmatrix} R \\ 0 \end{bmatrix} s. \tag{6.5}$$

Introduce λ to denote the mod-squared entries of z.

$$\lambda_i = |z_{N-i+1}|^2 \quad \text{for} \quad i = 1, \ldots, N.$$

Note that λ is indexed backwards relative to z. From (6.4), finding all s that satisfy (6.3) amounts to finding all s that satisfy

$$\lambda_1 + \lambda_2 + \ldots + \lambda_N \leq r^2. \tag{6.6}$$

Consider the lower $N - M$ entries of z. These are given by the vector $Q_2^* x$. Now, x is known to the receiver and since it knows H, it can calculate Q and R. Therefore $Q_2^* x = [z_{M+1}, \ldots, z_N]^T$ is known to the receiver. Hence, so are $\lambda_1, \ldots, \lambda_{N-M}$. Moreover, these are independent of s and $\tilde{s}$ and therefore contain no useful information for the decoder. Therefore, solving (6.6) is equivalent to solving

$$\lambda_{N-M+1} + \cdots + \lambda_N \leq r^{'2} \tag{6.7}$$

for $r^{'2} = r^2 - (\lambda_1 + \cdots + \lambda_{N-M})$. Note that due to the upper-triangularity of R, λ_{i+N-M} depends only on the unknowns $s_M, \ldots, s_{N-i+1}$ for $i = 1, \ldots, M$. Therefore (6.7) can be solved by successively solving

$$\begin{aligned} \lambda_{1+N-M} &\leq r^{'2}, \\ \lambda_{1+N-M} + \lambda_{2+N-M} &\leq r^{'2}, \\ &\vdots \\ \lambda_{1+N-M} + \lambda_{2+N-M} + \cdots + \lambda_N &\leq r^{'2}, \end{aligned} \tag{6.8}$$

for $s_M, s_{M-1}, \ldots, s_1$. This works in the following way. The first condition gives possible values for s_M. For each of these, using the second condition, we obtain possible val-

ues for s_{M-1}. This process continues because for any predetermined $s_M, \ldots, s_{M-i+2}$, the ith condition gives an interval for s_{M-i+1}. Once all $s \in \mathcal{S}^{M\times 1}$ that satisfy (6.7) are known, we can find that s which minimizes $\|x - Hs\|^2$. If there are no solutions found, we increase r' and resolve the problem. For more on the sphere decoder see [90].

6.4 Computational Complexity

Computational complexity is defined as the number of arithmetical operations required before the decoder gives an output. Apart from the complexity of the QR factorization, the major computation involved in finding the closest point is in determining all points in each lower dimension, i.e., in the successive inequalities of (6.8). We see that the algorithm constructs a search tree where the branches at depth k in the tree correspond to the lattice points inside the hypersphere of radius r and dimension k. Clearly, the total computation involved depends on the *number* of points the decoder visits as it constructs the tree. For a point in the kth dimension, the number of operations or flops required to process it turn out to be proportional to k. ($2k + 17$ in [90].) Therefore we have

$$C = \sum_{k=1}^{M} (\text{Expected \# of points in } k\text{-sphere of radius } r) \cdot (\text{flops/point}). \quad (6.9)$$

Thus, the complexity of the algorithm depends on the *size* of the search tree and the computation required at each dimension. For various implementations the flops/point can take different values and have a complicated dependence on the enumeration method especially for hardware implementations[102]. In particular, the pseudocode of [90] and that presented in Section 6.5.3 use a number of flops linear in the dimension under consideration. We will see in the analyses presented in this chapter that this factor either plays no role (asymptotic analysis of Section 6.7.2) or remains transparent in the final expression (Sections 6.7.1 and 6.7.3). Thus, replacing it by a different expression presents no difficulty as far as the analysis is concerned. In the

simulations, our particular implementation does use flops/point that are linear in the dimension and we use that fact while presenting numerical results.

For the setup involving a real channel and 2-PAM as the signal space and with the receiver using sphere decoding, [90] obtains the following complexity:

$$C = \sum_{k=1}^{M} (2k + 17) \sum_{l=0}^{k} \binom{k}{l} \Gamma\left(\frac{r^2}{2(1 + \frac{4\rho}{M} l)}, \frac{k + N - M}{2} \right) \tag{6.10}$$

where $\Gamma(x, a) = \int_0^x \frac{e^{-t}}{\Gamma(a)} t^{a-1} dt$ is the incomplete gamma function. [90] also has similar expressions for other constellations.

While the sphere decoding algorithm is one of the exact methods that solve the maximum-likelihood problem without exhaustive search, even with finite constellations (L-PAM, L^2-QAM, etc.), it begins to take up significantly more than N^3 or N^4 computations at some N that is in the range of practical interest. The reason for this is understood as follows. The chosen radius-squared, r^2, is typically proportional to N, therefore the algorithm retains a very large fraction of the lattice points (in fact nearly all the points) upto some dimension k before it starts to prune the tree. For instance, if $N = 100$, we have $r^2 = \alpha N$ such that up to dimension $k = cN$ where c is some constant less than 1, we keep nearly all the points of the lattice. This already gives us L^{cN} points to search over and the complexity quickly becomes exponential. The result of [91] makes this observation rigorous and we will discuss this issue further in Section 6.7.2.

6.5 Statistical Pruning

With a view to decreasing the computational complexity we now propose a modification to the sphere decoding algorithm that reduces the size of the tree. We suggest the Increasing Radii Algorithm, which defines a region around x, different from the hypersphere, to search in. This algorithm does not perform exact ML decoding but can perform as close to ML as desired through the choice of certain parameters. The proposed algorithm relies heavily on the statistics of the problem (such as the distri-

bution of the λ_i) for performance as well as reduction in complexity. In fact, it is the statistics that motivate the particular pruning approach that we take.

6.5.1 Statistics

We now take a look at these statistics. For any vector $s \in \mathcal{S}^{M\times 1}$ define $s^i \in \mathcal{S}^{i\times 1}$ as the lower length-i subvector of s, i.e., the vector $[s_{M-i+1}, \ldots, s_M]^T$. Define $c_i = \frac{1}{\sigma_v^2+\sigma_h^2\|s^i-\tilde{s}^i\|^2}$ and $c_0 = \frac{1}{\sigma_v^2} = 1$.

The characteristic functions and distributions for the λ_i random variables are obtained in Appendix 6.10.1 and mentioned in Table 6.1. The mean and variance can then be computed easily and are mentioned in Table 6.2.

	$Ee^{j\alpha\lambda_i}$	$p_{\lambda_i}(\lambda_i)$
$i \le N-M$	$\frac{1}{1-\frac{j\alpha}{c_0}}$	$c_o e^{-c_0\lambda_i}$
$i > N-M$	$\frac{(1-\frac{j\alpha}{c_{i-1-N+M}})^{i-1}}{(1-\frac{j\alpha}{c_{i-N+M}})^{i}}$	$\frac{c^i_{i-N+M}}{c^{i-1}_{i-1-N+M}} e^{-c_{i-N+M}\lambda_i} \sum_{k=0}^{i-1} \binom{i-1}{k} \frac{\lambda_i^k}{k!} (c_{i-1-N+M} - c_{i-N+M})^k$

Table 6.1: Characteristic function and pdf of λ_i

	$E\lambda_i$	var λ_i
$i \le N-M$	$\frac{1}{c_0}$	$\frac{1}{c_0^2}$
$i > N-M$	$\frac{i}{c_{i-N+M}} - \frac{(i-1)}{c_{i-1-N+M}}$	$\frac{i}{c^2_{i-N+M}} - \frac{(i-1)}{c^2_{i-1-N+M}}$

Table 6.2: Mean and variance of λ_i

We note that the λ_is are independent random variables. Define $\beta_{i,j} = \sum_{k=i}^{j} \lambda_{k+N-M}$ for $1 \le i \le j \le M$. We denote $\beta_{1,i}$ by β_i. Thus, β_i is simply the sum of i independent random variables. Therefore, its characteristic function is the product of the relevant λ_j characteristic functions. Now the statistics for the β_i random variables are easy to compute and are shown in Table 6.3. Note that the β_i are the quantities on the left side of (6.8).

The sphere decoder gives exponential complexity because the first several conditions of (6.8) are very loose. Thus, the tree of the points visited grows exponentially for the first several dimensions. This is also clear from the fact that the sums

$E^{j\alpha\beta_i}$	$p_{\beta_i}(\beta_i)$	$E\beta_i$	var β_i
$\frac{(1-\frac{j\alpha}{c_0})^{N-M}}{(1-\frac{j\alpha}{c_i})^{i+N-M}}$	$\frac{c_i^{i+N-M}}{c_0^{N-M}} e^{-c_i\beta_{1,i}} \sum_{l=0}^{N-M} \binom{N-M}{l} \frac{\beta_{1,i}^{i+l-1}}{(i+l-1)!}(c_0-c_i)^l$	$\frac{i+N-M}{c_i} - \frac{N-M}{c_0}$	$\frac{i+N-M}{c_i^2} - \frac{N-M}{c_0^2}$

Table 6.3: Statistics of β_i, $1 \leq i \leq M$

$\lambda_{1+N-M} + \cdots + \lambda_{k+N-M}$ which occur in (6.8) (viz., the β_ks) have monotonically increasing means while r' is typically chosen on the basis of the distribution of β_M, i.e., the full sum of all the λ_is under consideration. Therefore the first several conditions do not prune the search space as much as desired. Taking our cue from this, we propose a modification to the sphere decoding algorithm. In this modification, we prune the search space right from the lower dimensions.

6.5.2 Increasing Radii Algorithm (IRA)

Using a schedule of radii $r_1 \leq r_2 \leq \ldots \leq r_M$ we solve for

$$\begin{aligned} \lambda_{1+N-M} &\leq r_1^2, \\ \lambda_{1+N-M} + \lambda_{2+N-M} &\leq r_2^2, \\ &\vdots \\ \lambda_{1+N-M} + \lambda_{2+N-M} + \cdots + \lambda_N &\leq r_M^2, \end{aligned} \tag{6.11}$$

instead of solving for (6.8). By choosing a smaller radius for the lower dimensions and gradually increasing it, the search space is cut down much earlier than with the sphere decoder. We hope that this will reduce the number of points in the search region at the lower dimensions. Denote by $\mathcal{D}_k$ the region in $\mathcal{S}^{k\times 1}$ containing points that satisfy the first k inequalities of (6.11). (Note that these points have been determined by finding the values of s_M, s_{M-1}, ..., s_{M-k+1} that satisfy the first k conditions.) We refer to $\mathcal{D}_M$ as $\mathcal{D}$ in the following discussion. As in the sphere decoder we can determine all $s \in \mathcal{D}$ by solving the inequalities in (6.11) successively. Once the points within $\mathcal{D}$ are determined, we find that point in $\mathcal{D}$ which minimizes $\|x - Hs\|$ and declare it as the decoder output.

To reduce the complexity, we naturally try to reduce the number of points in $\mathcal{D}$. However, because of the "asymmetry" of the region it is possible that the lattice point closest to x does not lie in the search space. For the sphere decoder, the closest point to x inside the hypersphere is the closest point to x in the entire lattice. For the IRA, however, the closest point to x in $\mathcal{D}$ is not necessarily the closest point to x in the entire lattice. Thus, unlike the sphere decoder, we are *not* doing ML decoding and are, potentially, incurring a greater BER. What we get in return is reduced computational complexity. By increasing the asymmetry of the search region we can decrease the computation involved, but simultaneously incur an increased BER. This is the tradeoff inherent in the modification. As with the sphere decoder, if $\mathcal{D}$ is empty, we increase the search region and run the decoder again. We note in passing that similarly named algorithms are presented in [95]. However, they differ significantly from this method of pruning as they rank most promising paths within a fixed radius search in order to limit computation.

6.5.3 Pseudocode

The algorithm is in pseudocode in Table 6.4. It uses a depth-first search to construct the tree. We use the vector r of size $M \times 1$ to denote the schedule $r_1, \ldots, r_M$ that we are using for the decoding. GetNewSchedule returns the new sequence of r_is with which we repeat the search when the region $\mathcal{D}$ is empty. The first schedule is chosen so as to be successful with some probability $(1 - \epsilon_1)$. If it fails, the second is chosen so as to be successful with probability $(1 - \epsilon_2)$ etc. This will become clearer in later sections.

Clearly, for all r_i being equal the IRA is the same as the sphere decoder.

6.6 Probability of Error

The algorithm repeats the search with a new sequence of r_is if the solution set of (6.11), viz. $\mathcal{D}$, is empty. Let $\mathcal{D}^i$ be the solution set at the ith iteration. The algorithm terminates at the first i for which $\mathcal{D}^i \neq \emptyset$. We assume that $\mathcal{D}^{i-1} \subseteq \mathcal{D}^i$

and $\mathcal{D}^\infty = \mathcal{S}^{M\times 1}$.

Recall that $\tilde{s}$ is the transmitted point. Define $\epsilon_i = P(\tilde{s} \notin \mathcal{D}^i)$. With probability $P(\text{error})$ we make an error by decoding to $s \neq \tilde{s}$.

$$
\begin{aligned}
& P(\text{error}) \\
= & \sum_{i=1}^{\infty} P(\text{error}, \mathcal{D}^i \neq \emptyset, \mathcal{D}^{i-1} = \emptyset) \\
= & \sum_{i=1}^{\infty} P(\text{error}, \mathcal{D}^i \neq \emptyset, \mathcal{D}^{i-1} = \emptyset, \tilde{s} \in \mathcal{D}^i) + \sum_{i=1}^{\infty} P(\text{error}, \mathcal{D}^i \neq \emptyset, \mathcal{D}^{i-1} = \emptyset, \tilde{s} \notin \mathcal{D}^i) \\
= & \sum_{i=1}^{\infty} P(\|x - H(s-\tilde{s})\|^2 \leq \|v\|^2 \text{for } s \in \mathcal{D}^i, s \neq \tilde{s}, \mathcal{D}^i \neq \emptyset, \mathcal{D}^{i-1} = \emptyset, \tilde{s} \in \mathcal{D}^i) \\
& \qquad + \sum_{i=1}^{\infty} P(\mathcal{D}^i \neq \emptyset, \mathcal{D}^{i-1} = \emptyset, \tilde{s} \notin \mathcal{D}^i) \\
\leq & \sum_{i=1}^{\infty} P(\text{ML decoder error}, \mathcal{D}^i \neq \emptyset, \mathcal{D}^{i-1} = \emptyset) + \sum_{i=1}^{\infty} P(\mathcal{D}^i \neq \emptyset, \mathcal{D}^{i-1} = \emptyset, \tilde{s} \notin D^i) \\
= & P_e^{ML} + \sum_{i=1}^{\infty} P(\mathcal{D}^i \neq \emptyset, \mathcal{D}^{i-1} = \emptyset, \tilde{s} \notin \mathcal{D}^i) \qquad (6.12) \\
\leq & P_e^{ML} + \sum_{i=1}^{\infty} P(\mathcal{D}^i \neq \emptyset, \mathcal{D}^{i-1} = \emptyset, \tilde{s} \notin \mathcal{D}^1) \\
= & P_e^{ML} + \epsilon_1 \qquad (6.13)
\end{aligned}
$$

where P_e^{ML} is the probability of error with ML decoding. The third equality comes from the fact that an error is certain to be made if $\mathcal{D}^i \neq \emptyset, \mathcal{D}^{i-1} = \emptyset, \tilde{s} \notin \mathcal{D}^i$ since the transmitted point is not in $\mathcal{D}^i$ while some other point is. The first inequality comes from the fact that an ML decoder error does not require s or $\tilde{s}$ to be $\mathcal{D}^i$. We expect that (6.12) is a tight bound relating the probability of error of the modified algorithms to P_e^{ML}. This is because it takes into account all the successive schedules of r_i that the algorithms may go through. However it is not clear how to evaluate it exactly and hence we propose the simple bound of (6.13). This would be equal to (6.12) if we chose to use only one schedule of r_i and declared all bits to be in error if the corresponding $\mathcal{D}$ turned out to be empty, rather than increasing the r_i and running the decoder again.

6.6.1 ϵ with Increasing Radii Algorithm

For any given set of radii $r_1 \leq \ldots \leq r_M$, $\mathcal{D}$ denotes the set of the lattice points inside the search region. We now compute $\epsilon = P(\tilde{s} \notin \mathcal{D})$ for the Increasing Radii Algorithm.

Lemma 6.1. *For the IRA, given a set of radii $r_1 \leq \ldots \leq r_M$, $\epsilon = P(\tilde{s} \notin \mathcal{D})$ is given by*

$$\epsilon = \sum_{k=1}^{M} e^{-r_k^2} J_{k-1} \tag{6.14}$$

where

$$J_k = \sum_{l=0}^{k-1} (-1)^{k-l+1} \frac{r_{l+1}^{2(k-l)}}{(k-l)!} J_l, \qquad J_0 = 1 \tag{6.15}$$

Proof. If $s = \tilde{s}$, we have $z = Q^* v$. Since Q is unitary, $Q^* v$ has the same statistics as v, i.e., i.i.d. entries distributed as $\mathbb{CN}(0,1)$. With $\lambda_i = |z_{N-i+1}|^2$, we have $p_{\lambda_i}(\lambda_i) = e^{-\lambda_i}$. $1-\epsilon$ is the probability that $\lambda_{1+N-M}, \ldots, \lambda_N$ satisfy (6.11). Because the λ_is are independent,

$$p_{\lambda_{1+N-M},\lambda_{2+N-M},\ldots,\lambda_N}(\lambda_{1+N-M}, \lambda_{2+N-M}, \ldots, \lambda_N) = e^{-(\lambda_{1+N-M}+\lambda_{2+N-M}+\cdots+\lambda_N)}$$

Therefore

$$\begin{aligned} 1-\epsilon &= \int_0^{r_1^2} \int_0^{r_2^2-\lambda_{1+N-M}} \cdots \int_0^{r_M^2-(\lambda_{1+N-M}+\cdots+\lambda_{N-1})} e^{-(\lambda_{1+N-M}+\cdots+\lambda_N)} d\lambda_N \ldots d\lambda_{1+N-M} \\ &= \int_0^{r_1^2} \int_{\mu_1}^{r_2^2} \cdots \int_{\mu_{M-1}}^{r_M^2} e^{-\mu_M} d\mu_M \ldots d\mu_1 \end{aligned}$$

where the second line comes from changing variables: $\mu_i = \sum_{j=1}^{i} \lambda_{j+N-M}$ for $i = 1, \ldots, M$. If we call this integral I_M and integrate out μ_M we get

$$I_M = I_{M-1} - e^{-r_M^2} J_{M-1} \tag{6.16}$$

where $J_{M-1} = \int_0^{r_1^2} \int_{\mu_1}^{r_2^2} \cdots \int_{\mu_{M-2}}^{r_{M-1}^2} d\mu_{M-1} \ldots d\mu_1$. It can be shown that the J_is satisfy the recurrence of (6.15). Thus $J_0, \ldots, J_{M-1}$ can be computed. We define $I_0 = 1$. Then, using (6.16) recursively, we get $I_M = 1 - \sum_{k=1}^{M} e^{-r_k^2} J_{k-1}$. Since $1-\epsilon = I_M$, we

get (6.14). □

6.6.2 Choice of ϵ and the Radii

Thus we obtain an exact expression for ϵ. Once we decide how much worse than ML we are prepared to be, we can choose ϵ using the bound in (6.13). As indicated earlier, this bound is loose, and the performance is usually much better than that indicated by the value of ϵ. For the chosen value of ϵ, we can then use the expressions above to determine the radii $r_1, \ldots, r_M$. Note, however, that since (6.14) gives a highly underdetermined equation system involving the r_is there is an entire family of schedules of r_i that give a particular epsilon. But if we choose a functional form for the radii we can use the expressions obtained above to determine the r_is. Since we want to solve (6.11), choosing the r_is in accordance with the expected values of the partial sums that appear on the left side of each inequality is a reasonable option. But these partial sums are precisely the β_is. The statistics of these are in Table 6.3. We can see that their expected values are $\frac{i+N-M}{c_i} - \frac{N-M}{c_0} = (i+N-M)(\sigma_v^2+\sigma_h^2\|s^i-\tilde{s}^i\|^2)-(N-M)\sigma_v^2$. Although $\|s^i - \tilde{s}^i\|^2$ can take a range of values, we can see that $E\beta_i$ increases at least linearly with i. This motivates us to settle upon a linear schedule for the r_i^2s. This also means we have fewer parameters to choose. As indicated in the calculation of ϵ, the r_i^2 values are chosen with the noise statistics in mind, therefore the slope of linearity is chosen as σ_v^2. (This is typically one.) It is now enough to choose the value of r_1^2 to determine the entire schedule. If we choose $r_1^2 = (\delta \log M + 1)\sigma_v^2$, then the probability that the transmitted signal falls outside the search region at the first dimension decays as $\frac{1}{eM^\delta}$. Therefore we set $r_i^2 = (\delta \log M + i)\sigma_v^2$, and choose δ such that $\epsilon = 0.01$, etc. Thus we can stay as close to the ML performance as we desire through choice of r_is.

In Table 6.5 we list some values of δ for different values of M. This means that if we desire a value of ϵ for a particular value of M, a radius schedule of $r_i^2 = (\delta \log M + i)\sigma_v^2$ where δ is picked from the table will do the job.

6.7 Analysis of Computational Complexity

Recall the concept of computational complexity outlined in Section (6.4). In particular, we focus on the expression in (6.9). Since we are not searching over hyperspheres anymore we have a modified expression for the complexity.

$$C = \sum_{k=1}^{M} (\text{Expected \# of points in } \mathcal{D}_k) \cdot (\text{flops/point}). \tag{6.17}$$

From the pseudocode of Section 6.5.3 we can determine that the flops/point is $8k+32$.

Let us now investigate the exact computational complexity as defined in equation (6.17). s^k is as defined in Section 6.5.1. Define $P(s^k \in \mathcal{D}_k)$ to be the probability that the point s^k is in the search region at dimension k, i.e., it satisfies the first k equations of (6.11). Clearly

$$\text{Expected \# of points in } \mathcal{D}_k = \sum_{s^k \in \mathcal{S}^{k\times 1}} P(s^k \in \mathcal{D}_k). \tag{6.18}$$

We now need to compute $P(s^k \in \mathcal{D}_k)$ and then do the sum in (6.18). Note that the number of terms in the sum is L^{2k}, i.e., exponential in k. Naturally, we would like to evaluate the sum *without* having to explicitly evaluate $P(s^k \in \mathcal{D}_k)$ for each of the L^{2k} values of s^k. Whether this can be done or not depends on the functional form of $P(s^k \in \mathcal{D}_k)$. Therefore while determining $P(s^k \in \mathcal{D}_k)$ we also keep in mind the summation of (6.18).

For any $s^k \in \mathcal{S}^{k\times 1}$, the joint distribution of $\lambda_{1+N-M}, \ldots, \lambda_{k+N-M}$ determines $P(s^k \in \mathcal{D}_k)$. More specifically,

$$P(s^k \in \mathcal{D}_k) = \int_0^{r_1^2} \cdots \int_0^{r_k^2 - (\lambda_{1+N-M} + \cdots + \lambda_{k-1+N-M})} p_{\lambda_{1+N-M},\ldots,\lambda_{k+N-M}}(\lambda_{1+N-M}, \ldots, \lambda_{k+N-M}) d\lambda_{k+N-M} \ldots d\lambda_{1+N-M} \tag{6.19}$$

We know the distribution of the λ_is from Table (6.1). Since the λ_is are independent

we have

$$p_{\lambda_{1+N-M},\ldots,\lambda_{k+N-M}}(\lambda_{1+N-M},\ldots,\lambda_{k+N-M}) = \prod_{i=1}^{k} p_{\lambda_{i+N-M}}(\lambda_{i+N-M}) \tag{6.20}$$

Substituting from Table 6.1 and (6.20) into (6.19), the integral for $P(s^k \in \mathcal{D}_k)$ can be obtained exactly. However, this integral is very involved, and, moreover, even if evaluated exactly, would not give an expression that can be summed easily in (6.18). Therefore we now present one upper bound and one approximation to $P(s^k \in \mathcal{D}_k)$ and hence the complexity. We will also present an asymptotic analysis of the upper bound for large dimensions.

6.7.1 A Simple Upper Bound

We upper bound the number of points in the search region at dimension k by ignoring the fact that pruning has been done in dimensions less than k. This means that instead of imposing the first k conditions of (6.11) for a point to be in the search region at the kth subdimension, we only impose the kth condition. This becomes clearer in the proof of the following result.

Theorem 6.2. *For the Increasing Radii Algorithm the computational complexity is bounded as*

$$C \leq \sum_{k=1}^{M}(8k+32)\sum_{n=0}^{2k(L-1)^2} G_{L,k}[n]\sum_{l=0}^{N-M}\binom{N-M}{l}\left(\frac{1}{\sigma_v^2}-\frac{1}{\sigma_v^2+\sigma_h^2 n}\right)^l \times \sigma_v^{2(N-M)}(\sigma_v^2+\sigma_h^2 n)^{l-N+M}\Gamma\left(\frac{r_k^2}{\sigma_v^2+\sigma_h^2 n}, k+l\right) \tag{6.21}$$

where $G_{L,k}[n]$ is the coefficient of x^n in $\frac{1}{L^{2k}}\left(L+\sum_{j=1}^{L-1}2(L-j)x^{j^2}\right)^{2k}$ and $\Gamma(x,a) = \int_0^x \frac{e^{-t}}{\Gamma(a)}t^{a-1}dt$

Proof. Recall that $\beta_{i,j} = \sum_{k=i}^{j}\lambda_{k+N-M}$. For any s, let B_i be the event that $\beta_{1,i} \leq r_i^2$ for $i = 1,\ldots,M$. The statistics of the β_is are mentioned in Table 6.3. $s^k \in \mathcal{D}_k$ if it satisfies the first k conditions of (6.11). This happens with probability $P(B_1,\ldots,B_k)$.

Now, if we only wanted to impose the kth condition, it would be satisfied with probability $P(B_k)$. Naturally, $P(B_k)$ upperbounds $P(B_1, \ldots, B_k)$. Therefore,

$$\begin{aligned}
& P(s^k \in \mathcal{D}_k) \\
= \; & P(B_1, \ldots, B_k) \\
\leq \; & P(B_k) \\
= \; & \int_0^{r_k^2} p_{\beta_{1,k}}(\beta_{1,k}) d\beta_{1,k} \\
= \; & \sum_{l=0}^{N-M} \binom{N-M}{l} \frac{(c_0 - c_k)^l}{c_0^{N-M}} c_i^{N-M-l} \Gamma(c_k r_k^2, k+l) \\
= \; & \sum_{l=0}^{N-M} \binom{N-M}{l} \left(\frac{1}{\sigma_v^2} - \frac{1}{\sigma_v^2 + \sigma_h^2 \|s^k - \tilde{s}^k\|^2} \right)^l \sigma_v^{2(N-M)} (\sigma_v^2 + \sigma_h^2 \|s^k - \tilde{s}^k\|^2)^{l-N+M} \times \\
& \qquad \Gamma\left(\frac{r_k^2}{\sigma_v^2 + \sigma_h^2 \|s^k - \tilde{s}^k\|^2}, k+l \right)
\end{aligned}$$

where $\Gamma(x, a) = \int_0^x \frac{e^{-t}}{\Gamma(a)} t^{a-1} dt$ is the incomplete gamma function.

We now need to evaluate the summation of (6.18) with this upper bound. From the definition of $\mathcal{S}$ in (6.2) it is evident that each entry in $s^k - \tilde{s}^k$ can only take values of the form $x + jy$ where $x, y \in \{-(L-1), -(L-2), \ldots, (L-2), (L-1)\}$. Therefore $\|s^k - \tilde{s}^k\|^2$ can take values in $\{0, \ldots, 2k(L-1)^2\}$. Denote by $r_k^L(n)$ the "average" number of solutions to $\|s^k - \tilde{s}^k\|^2 = n$. More precisely

$$r_k^L(n) = \frac{1}{L^{2k}} \sum_{\tilde{s}^k \in \mathcal{S}^{k\times 1}} (\text{number of } s^k \in \mathcal{S}^{k\times 1} \text{ such that } \|s^k - \tilde{s}^k\|^2 = n) \tag{6.22}$$

We have assumed, without loss of generality, that all points are equally likely to be transmitted. With this the summation of (6.18) becomes

$$\begin{aligned}
& \sum_{s^k \in \mathcal{S}^{k\times 1}} P(s^k \in \mathcal{D}_{IR,k}) \qquad (6.23) \\
= \; & \sum_{n=0}^{2k(L-1)^2} r_k^L(n) \sum_{l=0}^{N-M} \binom{N-M}{l} \left(\frac{1}{\sigma_v^2} - \frac{1}{\sigma_v^2 + \sigma_h^2 n} \right)^l \sigma_v^{2(N-M)} (\sigma_v^2 + \sigma_h^2 n)^{l-N+M} \Gamma\left(\frac{r_k^2}{\sigma_v^2 + \sigma_h^2 n}, k+l \right).
\end{aligned}$$

It is shown in Appendix 6.10.2 that $r_k^L(n)$ is given by the coefficient of x^n in $G_L^k(x)$ where $G_L(x)$ is the generating function mentioned in the statement of Theorem 6.2. We denote $G_L^k(x)$ by $G_{L,k}(x)$ and the coefficient of x^n in this by $G_{L,k}[n]$. This gives us $r_k^L(n) = G_{L,k}[n]$. Using this in (6.23) and the expressions relating to complexity stated in (6.17) and (6.18), we get the upper bound in (6.21). □

This upper bound is very easy to evaluate, especially for small and moderate values of M and N. It is also quite tight in this region. We further note that for $N = M$, the upper bound of (6.21) simplifies to

$$C \leq \sum_{k=1}^{M} (8k + 32) \sum_{n=0}^{2k(L-1)^2} G_{L,k}[n] \Gamma \left(\frac{r_k^2}{\sigma_v^2 + \sigma_h^2 n}, k \right). \tag{6.24}$$

We also note that for the 4-QAM constellation, $L = 2$ and $G_{2,k}[n] = \binom{2k}{n}$.

The upper bound of this section is valid for all values of M, N, L, and SNR. In the following section, we fix $M = N$ and analyze this upper bound for a fixed SNR and asymptotically large N.

6.7.2 Asymptotics of the Upper Bound

In this section we will compare the asymptotic complexities of the sphere decoder and the upper bound on the Increasing Radii Algorithm using some simple arguments. We will assume $M = N$ and that N is very large. Let $r^2 = N$ for the sphere decoder and $r_i^2 = i$ for the IRA. (It turns out that having $r^2 = N + \delta \log N$ or $r_i^2 = i + \delta \log N$ for constant δ does not affect the asymptotic analysis.) The subscripts SD and IR will be used when we discuss the complexities of the sphere decoder and the IRA respectively. Although the analysis can be done for a generic QAM constellation, we only present results for 4-QAM. This is because the expression for $G_{L,k}[n]$ in the upperbound of the previous section is a simple binomial coefficient for this case and is more complicated in the generic case.

Consider the complexity expression for the sphere decoder for the case of $\mathcal{S}$ being the 4-QAM constellation. This is similar to that for the 2-PAM constellation given

in (6.10) except for the fact that at subdimension k, we are dealing with complex vectors of length k or real vectors of length $2k$. (This issue is addressed in [90].) We have the following expression:

$$C_{\mathrm{SD}} = \sum_{k=1}^{N}(8k+32)\sum_{l=0}^{2k}\binom{2k}{l}\Gamma\left(\frac{r^2}{1+\frac{2\rho}{N}l}, k\right) \tag{6.25}$$

where $\Gamma(x,a) = \int_0^x \frac{e^{-t}}{\Gamma(a)}t^{a-1}dt$. From (6.24) and since $G_{2,k}[n] = \binom{2k}{n}$, we have

$$C_{\mathrm{IR}} \leq U_{\mathrm{IR}} = \sum_{k=1}^{N}(8k+32)\sum_{l=0}^{2k}\binom{2k}{l}\Gamma\left(\frac{r_k^2}{1+\frac{2\rho}{N}l}, k\right) \tag{6.26}$$

Note that the only difference between (6.25) and (6.26) is that, within the incomplete Gamma function, the r^2 of the former is replaced by r_k^2 in the latter. We now compare C_{SD} and U_{IR}. Consider the following upper and lower bounds. Both expressions have $N(N+1)$ terms and the maximum value for $(8k+32)$ is $(8N+32)$. For large N we have $N(N+1)(8N+32) \leq 9N^3$. Therefore:

$$\max_{k=1,\ldots,N;\ l=0,\ldots,2k}\binom{2k}{l}\Gamma\left(\frac{r^2}{1+\frac{2\rho}{N}l}, k\right) \leq C_{\mathrm{SD}} \leq 9N^3 \max_{k=1,\ldots,N;\ l=0,\ldots,2k}\binom{2k}{l}\Gamma\left(\frac{r^2}{1+\frac{2\rho}{N}l}, k\right)$$

and

$$\max_{k=1,\ldots,N;\ l=0,\ldots,2k}\binom{2k}{l}\Gamma\left(\frac{r_k^2}{1+\frac{2\rho}{N}l}, k\right) \leq U_{\mathrm{IR}} \leq 9N^3 \max_{k=1,\ldots,N;\ l=0,\ldots,2k}\binom{2k}{l}\Gamma\left(\frac{r_k^2}{1+\frac{2\rho}{N}l}, k\right).$$

It is easy to show that

$$\Gamma(x,a) = e^{-x}\sum_{l=a}^{\infty}\frac{x^l}{l!} \geq e^{-x}\frac{x^a}{a!} \geq e^{-x}\frac{x^a}{\sqrt{2\pi a e}\left(\frac{a}{e}\right)^a} = \frac{e^{a-x}}{\sqrt{2\pi a e}}\left(\frac{x}{a}\right)^a$$

where the second inequality comes from Stirling's approximation for large a: $a! \leq$

$\sqrt{2\pi ae}\left(\frac{a}{e}\right)^a$. With this and since $k \leq N$, we have

$$\Gamma\left(\frac{\nu}{1+\frac{2\rho}{N}l}, k\right) \geq \frac{1}{\sqrt{2\pi ke}} \frac{e^{k-\frac{\nu}{1+\frac{2\rho}{N}l}}}{\left(\frac{k}{\nu}\left(1+\frac{2\rho l}{N}\right)\right)^k} \geq \frac{1}{\sqrt{2\pi Ne}} \frac{e^{k-\frac{\nu}{1+\frac{2\rho}{N}l}}}{\left(\frac{k}{\nu}\left(1+\frac{2\rho l}{N}\right)\right)^k}.$$

Now, if we upper bound $\Gamma\left(\frac{\nu}{1+\frac{2\rho}{N}l}, k\right)$ using a simple Chernoff bound, we get

$$\Gamma\left(\frac{\nu}{1+\frac{2\rho}{N}l}, k\right) \leq \frac{e^{k-\frac{\nu}{1+\frac{2\rho}{N}l}}}{\left(\frac{k}{\nu}\left(1+\frac{2\rho l}{N}\right)\right)^k} \qquad \text{for} \qquad k \geq \frac{\nu}{1+\frac{2\rho}{N}l}.$$

Note that the upper and lower bounds shown above differ only in the factor of $\frac{1}{\sqrt{2\pi Ne}}$.

Assume that $k = bN$ and $l = aN$ for constants a and b. Then $0 \leq b \leq 1$, $0 \leq a \leq 2b$. (The condition $k \geq \frac{\nu}{1+\frac{2\rho}{N}l}$ is always satisfied for the IRA since $\nu = r_k^2 = k$. For the sphere decoder, $\nu = r_k^2 = N$ and the condition translates to $b \geq \frac{1}{1+2\rho a}$.) The term $\binom{2k}{l}$ then looks like $\binom{2bN}{aN}$ and is equal to $\exp(2bNH(\frac{a}{2b}))$ for large N, where $H(p) = -p\log p - (1-p)\log(1-p)$ is the entropy function. This gives

$$\binom{2k}{l} \frac{e^{k-\frac{N}{1+\frac{2\rho}{N}l}}}{\left(\frac{k}{N}\left(1+\frac{2\rho l}{N}\right)\right)^k} \doteq \exp\left\{N\left(2bH(\frac{a}{2b}) + b - \frac{1}{1+2\rho a} - b\log(b(1+2\rho a))\right)\right\} = \exp(N\gamma_{\text{SD}}(a,b))$$

where we define $\gamma_{\text{SD}}(a,b) = 2bH(\frac{a}{2b}) + b - \frac{1}{1+2\rho a} - b\log(b(1+2\rho a))$. Also,

$$\binom{2k}{l} \frac{e^{k-\frac{k}{1+\frac{2\rho}{N}l}}}{\left(\frac{k}{k}\left(1+\frac{2\rho l}{N}\right)\right)^k} \doteq \exp\left\{N\left(2bH(\frac{a}{2b}) + \frac{2\rho ab}{1+2\rho a} - b\log(1+2\rho a)\right)\right\} = \exp(N\gamma_{\text{IR}}(a,b))$$

where we define $\gamma_{\text{IR}}(a,b) = 2bH(\frac{a}{2b}) + \frac{2\rho ab}{1+2\rho a} - b\log(1+2\rho a)$. Thus the bounds for C_{SD} become

$$\frac{1}{\sqrt{2\pi Ne}} \max_{0\leq b\leq 1, 0\leq a\leq 2b, b\geq\frac{1}{1+2\rho a}} \exp(N\gamma_{\text{SD}}(a,b)) \dot{\leq} C_{\text{SD}} \dot{\leq} 9N^3 \max_{0\leq b\leq 1, 0\leq a\leq 2b, b\geq\frac{1}{1+2\rho a}} \exp(N\gamma_{\text{SD}}(a,b))$$

and the bounds on U_{IR} become

$$\frac{1}{\sqrt{2\pi Ne}} \max_{0\le b\le 1, 0\le a\le 2b} \exp(N\gamma_{\mathrm{IR}}(a,b)) \dot{\le} U_{\mathrm{IR}} \dot{\le} 9N^3 \max_{0\le b\le 1, 0\le a\le 2b} \exp(N\gamma_{\mathrm{IR}}(a,b)).$$

It is easy to check that there are values of a and b for which $\gamma_{\mathrm{SD}}(a,b)$ and $\gamma_{\mathrm{IR}}(a,b)$ are positive, thus giving exponential bounds on the complexity. Therefore the terms $\frac{1}{\sqrt{2\pi Ne}}$ and $9N^3$ are asymptotically insignificant. Thus, the upper and lower bounds match and we have the exact asymptotic complexity behavior. If we denote the asymptotic complexities of the sphere decoder and the IRA by $e^{\gamma_{\mathrm{SD}}N}$ and $e^{\gamma_{\mathrm{IR}}N}$ respectively, we get

$$\gamma_{\mathrm{SD}} = \max_{0\le b\le 1, 0\le a\le 2b, b\ge \frac{1}{1+2\rho a}} \gamma_{\mathrm{SD}}(a,b) = \max_{0\le b\le 1, 0\le a\le 2b, b\ge \frac{1}{1+2\rho a}} 2bH(\frac{a}{2b}) + b - \frac{1}{1+2\rho a} - b\log(b(1+2\rho a))$$

and

$$\gamma_{\mathrm{IR}} = \max_{0\le b\le 1, 0\le a\le 2b} \gamma_{\mathrm{IR}}(a,b) = \max_{0\le b\le 1, 0\le a\le 2b} 2bH(\frac{a}{2b}) + \frac{2\rho ab}{1+2\rho a} - b\log(1+2\rho a).$$

Both maximizations are easy to perform numerically. In Figure 6.1 we plot the gamma values obtained from the maximizations for different SNRs. Not surprisingly, γ_{IR} is much lower than γ_{SD}. This means that the upper bound on the IRA is much lower than the complexity of the sphere decoder. This implies that the actual complexity of the IRA will be even less compared to the complexity of the sphere decoder.

We note in passing that although the large deviations approach of [91] is quite different, it gives exactly the same numerical results as the maximization for γ_{SD} above. Furthermore, using a similar large deviations approach for the asymptotic analysis of (6.26) leads to the same γ_{IR} as above.

6.7.3 Approximate Analysis

We now present an approximate method of computing the complexity that can be used for larger values of M and N where the upper bound may be harder to compute.

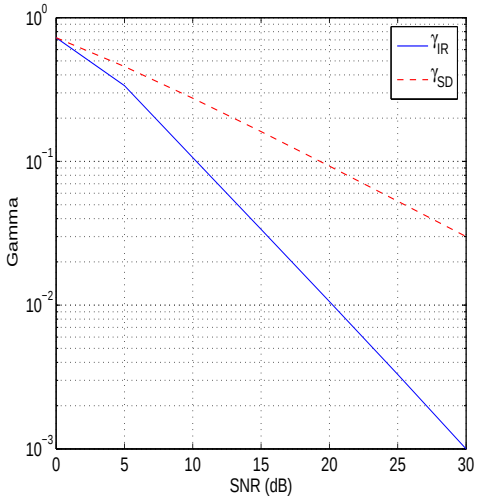


Figure 6.1: For large N, the complexities of the sphere decoder and the IRA are given by $e^{\gamma_{\rm SD}N}$ and $e^{\gamma_{\rm IR}N}$, respectively, where γ is as plotted. At 20 dB, $\gamma_{\rm SD}$ is roughly 10 times $\gamma_{\rm IR}$.

It is useful to think of the Increasing Radii Algorithm in the context of a random walk on the positive real line. If λ_{i+N-M} is the random variable denoting the length of the ith step, we can think of the sums $\lambda_{1+N-M} + \lambda_{2+N-M} + \cdots + \lambda_{i+N-M}$ as the total distance covered in the first i steps. In the Increasing Radii algorithm, we are interested in the joint probability that at the end of the ith step the distance covered is not more than r_i^2 *for all* $i = 1, \ldots, M$. Recall the definition of $\beta_{i,j}$. Note that since the λ_{i+N-M}s are independent (and not identical) and hence independent of $\beta_{1,i-1}$, a parallel interpretation for the sequence $\beta_{1,1}, \beta_{1,2}, \ldots, \beta_{1,M}$ is also that of a (non-stationary) Markov chain in discrete time and continuous space, i.e., $\beta_{1,1} \rightarrow \beta_{1,2} \rightarrow \ldots \rightarrow \beta_{1,M}$. More generally, if we start at some arbitrary step k, and stop at step $k+j$, $j \geq 0$ we still have a Markov chain given by $\beta_{k,k} \rightarrow \beta_{k,k+1} \rightarrow \ldots \rightarrow \beta_{k,k+j}$. Using this insight we can approximate $P(s^k \in \mathcal{D}_k)$ in a simpler manner rather than attempt the intractable integral of (6.19). We state this in the following theorem.

Theorem 6.3. *For any $s^k \in \mathcal{S}^{k\times 1}$, the probability that $s^k \in \mathcal{D}_k$ is given by*

$$P(s^k \in \mathcal{D}_k) \approx \prod_{i=1}^{k} \int_0^{X_i} p_{\lambda_{i+N-M}}(\lambda_{i+N-M}) d\lambda_{i+N-M} \tag{6.27}$$

where $X_1 = r_1^2$ *and* $X_2, \ldots, X_k$ *satisfy*

$$\int_0^{r_i^2 - r_{i-1}^2 - X_i + X_{i-1}} p_{\lambda_{i+N-M}}(\lambda_{i+N-M}) d\lambda_{i+N-M} = \frac{1}{2} \int_0^{X_{i-1}} p_{\lambda_{i+N-M}}(\lambda_{i+N-M}) d\lambda_{i+N-M}. \tag{6.28}$$

The approximation error in (6.27) is given in equation (6.37) of the Appendix.

Proof. Recall the preceding discussion regarding Markov chains. In order to use this point of view, we first propose the following approximation, proved in Appendix 6.10.3.1.

$$\int_0^{\Delta} f(x)g(x)dx = g(x^{'}) \int_0^{\Delta} f(x)dx + O\left(\left[\int_0^{\Delta} f(x)dx \right]^2 \right) \tag{6.29}$$

which holds for any $x^{'} \in [0, \Delta]$. This approximation is especially good when $x^{'}$ satisfies

$$\int_0^{x^{'}} f(t)dt = \frac{1}{2} \int_0^{\Delta} f(t)dt \tag{6.30}$$

since the error term then becomes $O\left(\left[\int_0^{\Delta} f(x)dx \right]^3 \right)$.

In Appendix 6.10.3.2 we make use of this approximation to obtain (6.27) and (6.28). □

The above theorem helps us convert a k-fold integral into the product of k simple integrals in (6.27). While (6.27) is always true (with equality) for *some* values of X_i (using a generalized mean value theorem), the importance of Theorem 6.3 is in obtaining good values of X_i as given by (6.28). Furthermore, in Appendix 6.10.3.1 and 6.10.3.2 it is shown that if we solve (6.28) approximately we still get an expression similar to (6.27) but with a different error term. One simple approximate solution to (6.28), obtained using Chernoff bounds, leads to the recursion

$$X_i = r_i^2 - r_{i-1}^2 - \frac{1 + \log 2}{2c_{i-1}} + \frac{X_{i-1}}{2} + \frac{1}{2c_{i-1}} \sqrt{(1 + \log 2 + c_{i-1}X_{i-1})^2 - 4c_{i-1}X_{i-1}}. \tag{6.31}$$

From the pdf of λ_i in Table (6.1) we see that $\int_0^{X_k} p_{\lambda_{k+N-M}}(\lambda_{k+N-M}) d\lambda_{k+N-M}$ can be evaluated exactly and will be a linear combination of some incomplete gamma

functions. A faster but approximate method of evaluating this integral is

$$\int_0^{X_i} p_{\lambda_i}(\lambda_i)d\lambda_i \approx (Q(\sqrt{2c_i}(\mu - \sqrt{\pi X_i}/2)) - Q(\sqrt{2c_i}(\mu + \sqrt{\pi X_i}/2)))^2 \qquad (6.32)$$

where $\mu = \sqrt{(i-1+N-M)\sigma_h^2|\tilde{s}_i - s_i|^2/2}$ and $Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-x^2/2}dx$ is the standard Q-function. Although the approximation error has not been rigorously analyzed, we have found this to be a good approximation. Using (6.27) and (6.28) (with or without the simplifications of (6.31) and (6.32)), we compute $P(s^k \in \mathcal{D}_k)$ approximately. Substituting for this in (6.18) and (6.17), we get an approximate value for the complexity. The summation over $s^k \in \mathcal{S}^{k\times 1}$ required in (6.18) is not easy to perform and hence we evaluate it by Monte Carlo simulations.

6.8 Simulations

In this section we present the results of simulations for different systems. Numerical results for the i.i.d. systems analyzed in the chapter are presented, as are simulations for a linear dispersion code. In all examples, we have $M = N$. We present a comparison of symbol error rates and complexities for the sphere decoder (with Schnorr-Euchner) and those of IRA, for different QAM constellations and values of N and SNR.

We note that since H and $\mathcal{S}$ are complex this amounts to solving $2N$ dimensional real problems. The computational complexity C is presented through the complexity exponent $C_E = \log C/\log N$. With this, a complexity exponent of C_E means that the complexity is N^{C_E}. (Clearly, C_E is different from the γ of Section 6.7.2.)

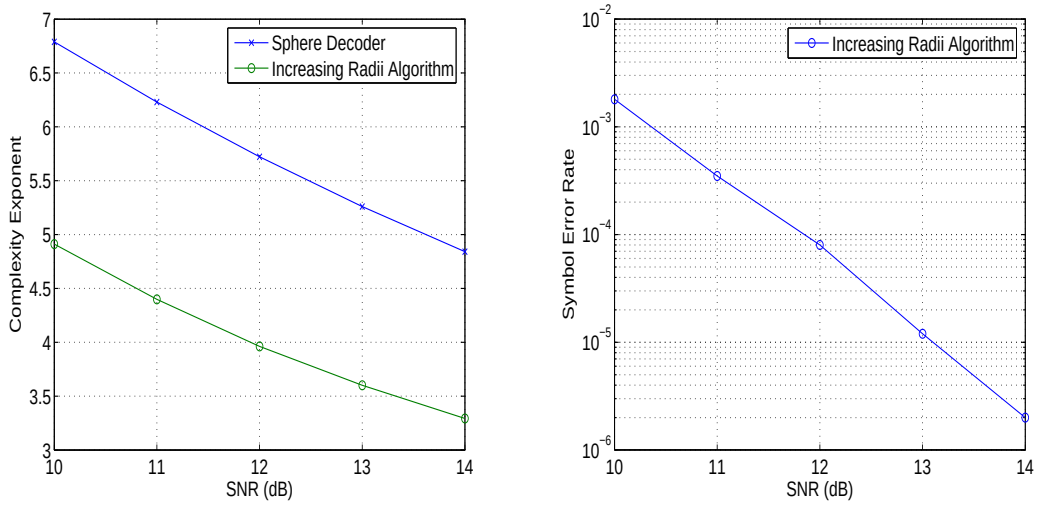
In all simulations for the sphere decoder we have used a value of r chosen to give a particular ϵ. For the Increasing Radii Algorithm we have used a linear schedule of radii, i.e., we have $r_i = i + \delta \log N$ where δ is chosen with some value of ϵ in mind. The sequence of ϵ_is that we use is simply 0.1, 0.01, 0.001, etc. This means that we first find r for the sphere decoder (δ for the IRA), which ensures that the transmitted vector is not in the search region with a probability of 0.1 and run the algorithm. If

the search region is empty we find a new value of r (δ for the IRA) that gives an ϵ of 0.01, and run the algorithm again. This continues till we find a non-empty search region.

Once we have at least one point in the search region, we find, from among those, that point s which minimizes $\|x - Hs\|^2$.

The expression in (6.17) is used to compute complexity where the (Expected # of points in $\mathcal{D}_k$) is estimated by running the decoder on many random instantiations of the problem. $(8k + 32)$ is the flops/point.

6.8.1 Computational Complexity and BER



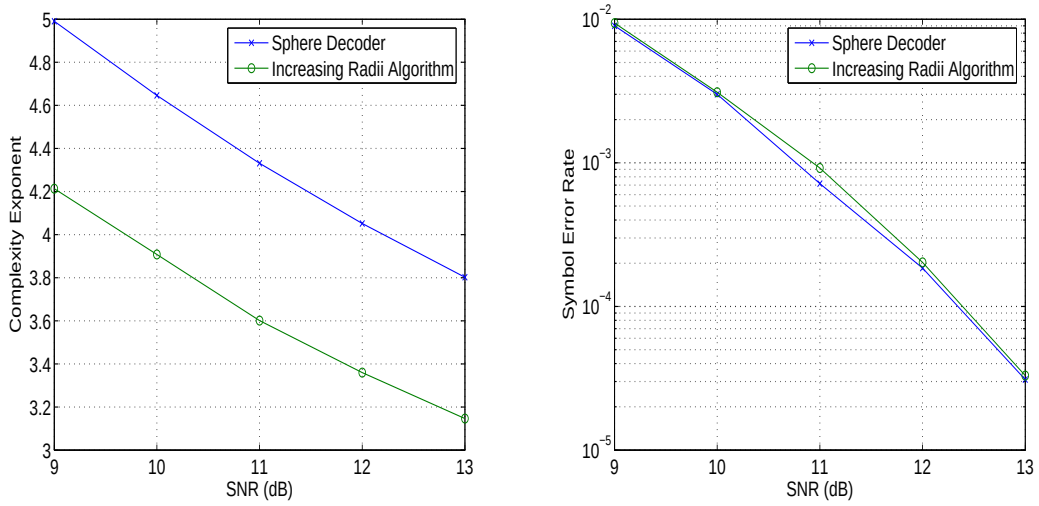
(a) Complexity Exponent vs. SNR.

(b) SER vs. SNR.

Figure 6.2: Complexity exponent and SER for $M = N = 50$ and 4-QAM. From Figure 6.2(a) we see that the IRA can be upto $50^{1.4} = 240$ times faster than the sphere decoder. Figure 6.2(b) shows the symbol error rate with the IRA.

In Figures 6.2, 6.3, and 6.4 we look at the complexity exponent and symbol error rate (SER) against the SNR for different values of N and and constellation size.

In Figure 6.2 we have $N = 50$ and $L = 2$, which is the 4-QAM constellation. The SNR ranges from 10 dB to 14 dB. In Figure 6.2(a), we see that the complexity



(a) Complexity Exponent vs. SNR.

(b) SER vs. SNR.

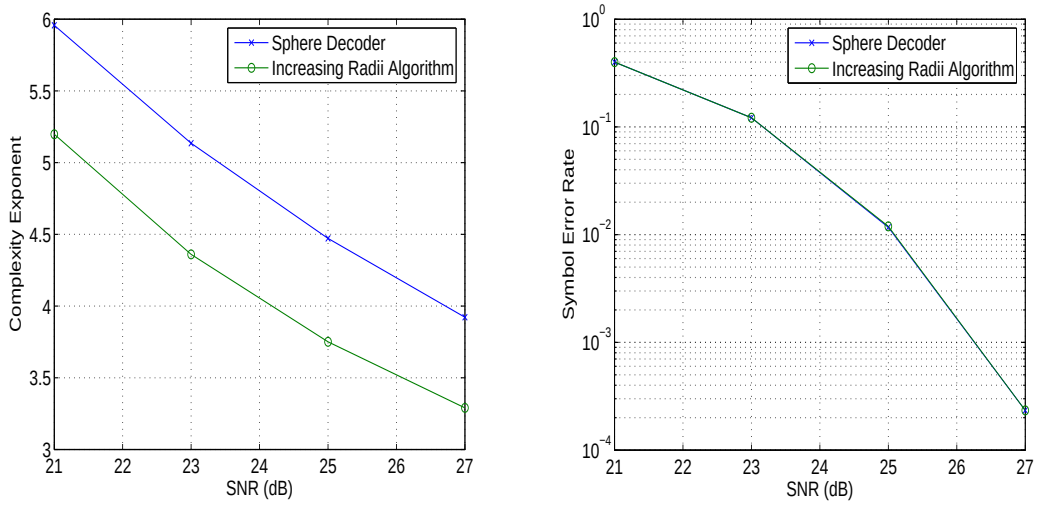
Figure 6.3: Complexity exponent and SER for $M = N = 20$ and 4-QAM. From Figure 6.3(a) we see that the IRA can be upto 11 times faster than the sphere decoder. From Figure 6.3(b) we see that the symbol error rates for the two algorithms are very close to each other, indicating no loss of performance.

exponent can be reduced significantly by using the IRA. We see a complexity that is up to 1.4 orders of magnitude smaller, which means that the IRA can run up to $50^{1.4} = 240$ times faster. In Figure 6.2(b), we see the SER for the IRA. Unfortunately, we have not been able to produce the SER plot for the sphere decoder for this dimension since it would take too long to obtain accurate values.

For the BER comparison we present results of a smaller sized problem, viz., N=20 in Figure 6.3. From Figure 6.3(a) and Figure 6.3(b) we see that with computational savings of 0.8 orders of magnitude (11 times less computation) we get a SER that is very close to the optimal SER ensured by ML decoding.

In Figure 6.4 we use $N = 12$ and $L = 8$. This corresponds to a 64-QAM constellation. From Figure 6.4(a) we see that the IRA runs around 7 times faster than the traditional sphere decoder. From the SER curves of Figure 6.4(b) we see that there is no loss of performance.

Not surprisingly, the savings from the IRA are more significant for large N. (This



(a) Complexity Exponent vs. SNR.

(b) SER vs. SNR.

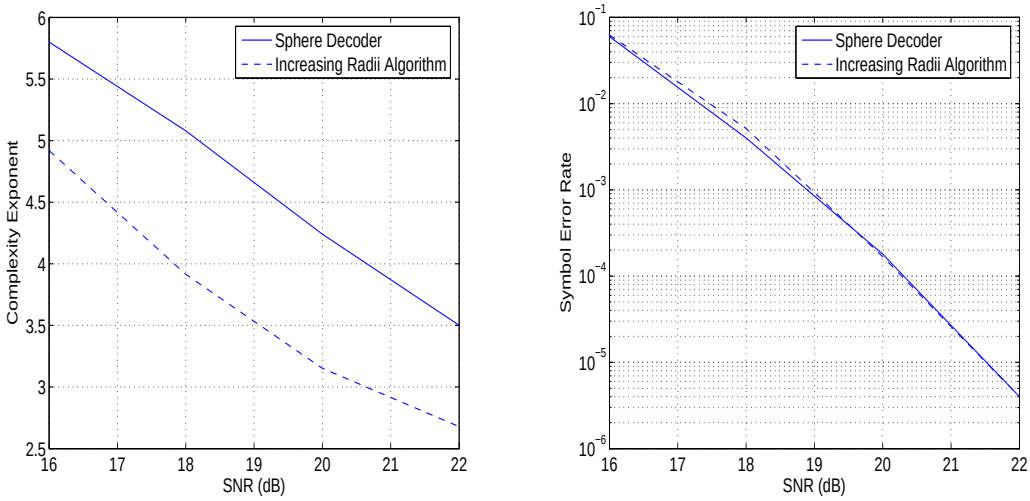
Figure 6.4: Complexity exponent and BER for $M = N = 12$ and 64-QAM. From Figure 6.4(a) we see that the IRA can be upto 7 times faster than the sphere decoder. From Figure 6.4(b) we see that the symbol error rates for the two algorithms are very close to each other, indicating no loss of performance.

will be further demonstrated in a later simulation.) In fact, for systems of dimension 6 and lower we find that the gains relative to the sphere decoder are minimal. This is because the pruning affects fewer dimensions and the overall complexity is unaffected. Another observation to make from the above set of plots is that (6.13) is a loose bound since for this setup it says that the proposed algorithms can give SERs that are as much as 0.1 above the optimal. The simulations indicate that this is a gross over-estimate.

6.8.2 Decoding in a Space-Time Coded System

In this section we consider the decoding of a system where the equivalent channel is given by a correlated H matrix rather than an i.i.d. one. Such systems arise commonly in space-time coded systems. We consider the linear dispersion code with eight transmit and four receive antennas with $T = 8$, $Q = 32$ and $R = 16$ presented in [79]. The constellation used is 16-QAM. The equivalent channel used for decoding

is a matrix of size 32×32 with correlated complex entries. Thus, the decoder works on a real system of dimension 64. In figure 6.5 we present curves for the complexity exponents and the symbol error rates for the sphere decoder (with Schnorr-Euchner) and the IRA. We see that the IRA is around 50 times faster and shows almost no loss in performance. Thus, the IRA presents a significant complexity savings while operating in space-time coded systems.

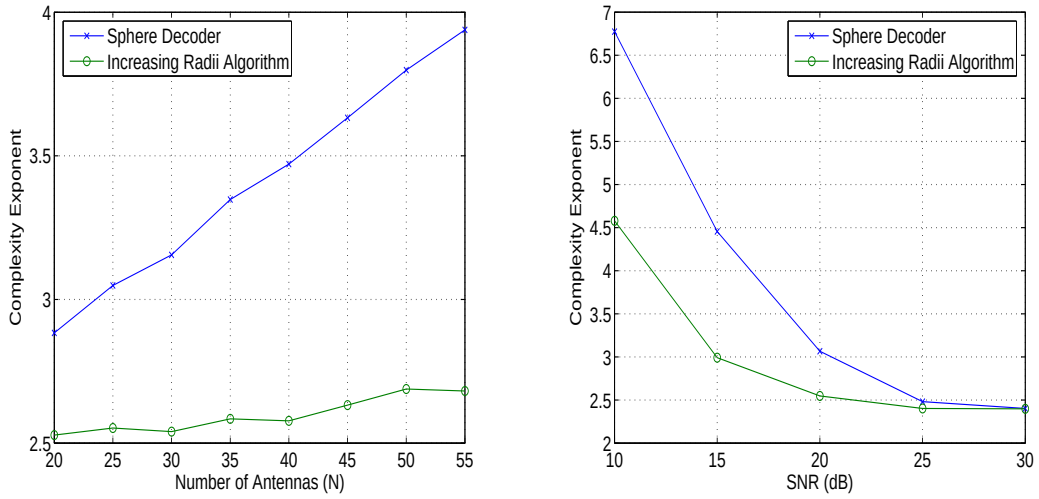


(a) Complexity Exponent vs. SNR.

(b) SER vs. SNR.

Figure 6.5: Complexity exponent and SER for the linear dispersion code with eight transmit and four receive antennas, with $T = 8$, $Q = 32$ and $R = 16$ with 16-QAM. From Figure 6.5(a) we see that the IRA is 50 times faster than the sphere decoder on average. From Figure 6.5(b) we see that the symbol error rates for the two algorithms are very close to each other, indicating no loss of performance.

Simulations for the smaller LD code in [79] with 4 transmit and 2 receive antennas and $T = 6$, $R = 8$, $Q = 4$ and 16-QAM were also done. This gave an equivalent channel of size 12×12. For this, the IRA ran roughly twice as fast as the sphere decoder with an identical symbol error rate in the SNR range of 15 dB to 25 dB.



(a) Dependence on N. SNR=27 dB, $L = 4$

(b) Dependence on SNR. $N = 50$, $L = 2$.

Figure 6.6: Dependence of complexity on N and SNR. Figure 6.6(a) plots the complexities of the two algorithms against the number of antennas, N. The complexity exponent of the sphere decoder increases much faster than that of the IRA. Figure 6.6(b) plots the two complexities against SNR. Computational savings with the IRA are more significant at low SNRs.

6.8.3 Comparing Complexities

From the previous section it is clear that the IRA can be used to give complexities that are much lower than that of the sphere decoder while still giving BERs close to optimal. Therefore in this section we only compare the complexities of the sphere decoder and the IRA.

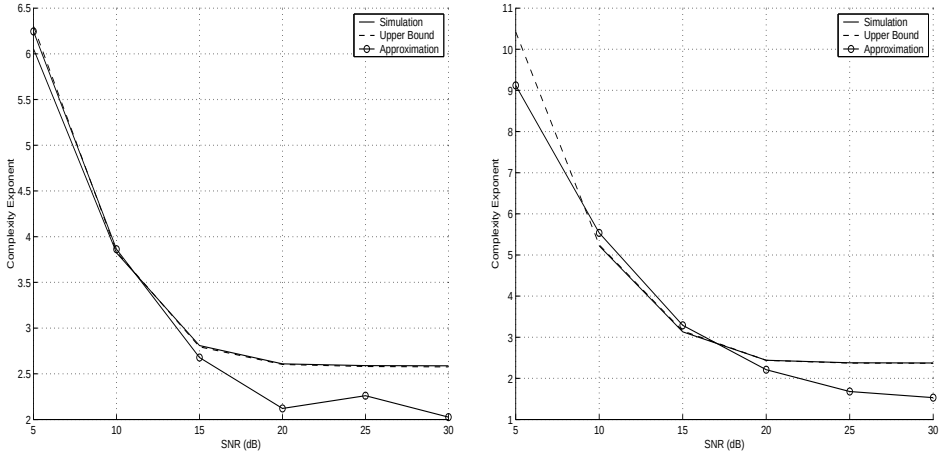
In Figure 6.6 we compare the complexity of the sphere decoder with that of the IRA in two different ways. In Figure 6.6(a), we set the SNR at 27 dB and $L = 4$, i.e., a 16-QAM constellation. We vary N from 20 to 55 and get estimates of the complexity by running the two algorithms sufficiently many times. We see that the complexity exponent of the sphere decoder is increasing rapidly while that of the IRA increases much more slowly. This bears out the analysis of Section 6.7.2 nicely.

In Figure 6.6(b), we set $N = 50$ and $L = 2$ (4-QAM constellation) and vary the

SNR from 10 dB to 30 dB. We see that the IRA consistently gives us a computational advantage, however, as the SNR increases, both decoders are quite fast and the relative advantage of the IRA diminishes. In particular, at 10 dB, we see that the IRA is around $50^{1.5} = 300$ times faster.

6.8.4 Simulations for the Upper Bound and Approximate Analysis for the IRA

We now compare the actual complexity of the Increasing Radii Algorithm as obtained by simulations, with the upper bound and approximation derived in Theorem 6.2 and Theorem 6.3.



(a) N=20, L=2. Complexity exponent vs. SNR.

(b) N=60, L=2. Complexity exponent vs. SNR.

Figure 6.7: Complexity exponent for the IRA – simulated, upper bound, and approximation. The simulations show that the complexity exponent for the IRA is tightly upper bounded by Theorem 6.2. The approximation of Theorem 6.3 is good up to SNRs of around 15 dB.

In Figure 6.7 we present curves that show the complexity exponent for the Increasing Radii Algorithm. For N being 20 and 60 and $L = 2$ (4-QAM constellation) and SNR ranging from 5 dB to 30 dB we show the complexity exponent obtained through simulation, by using the upper bound of Theorem 6.2 and the approxima-

tion of Theorem 6.3 with (6.32). We see that the upper bound is very good in this entire range. As for the approximation, we see that it is quite good for SNRs up to around 15 dB and then starts to underestimate the complexity.

6.9 Conclusions

This chapter examines the integer least-squares problem in a probabilistic setting, where the algorithmic complexity of decoding is a random variable. We use statistical pruning to reduce the search space – and therefore the complexity – while still keeping the transmitted point in the search region with high probability.

We have proposed a new method of doing this pruning and studied the complexity and the probability of error of the proposed method. Our proposed IRA algorithm achieves significant computational savings relative to the sphere decoder while still maintaining BERs close to optimal. For example, for a real problem in 100 dimensions, we observe a factor 240 savings in computation.

Many interesting questions remain to be answered. Finding an optimal schedule for the IRA seems to be quite challenging since our complexity expressions are not exact or analytically tractable. Simpler expressions for the complexity and BER might help better quantify the tradeoff between performance and complexity and also give insight into optimizing the radii schedules.

The sphere decoding technique can be used for joint detection and decoding of block codes [104]. By analogy, the modified algorithms are also applicable in this context. Analysis of performance and complexity in this scenario is interesting. Another potentially challenging question of interest, is how to choose the radii, given H. The smallest region around x that contains the closest point depends on H as well as v, but the current choice of r_i only takes the statistics of v into consideration.

We believe that the proposed pruning approach to the decoding problem demonstrates promise and that further work to analyze and optimize these statistical techniques will be of practical and theoretical interest.

6.10 Appendix

6.10.1 Derivation of Table (6.1)

For any $u \in \mathbb{C}^{T\times 1}$, we define $u^i = [u_{T-i+1}, \ldots, u_T]^T$. Consider the $H = Q \begin{bmatrix} R \\ 0 \end{bmatrix}$ decomposition where H is $N \times M$ with i.i.d. $\mathbb{CN}(0, \sigma_h^2)$ entries, Q is unitary of size $N \times N$ and R is upper triangular of size $M \times M$. It can be shown that the non-diagonal entries of R are i.i.d. $\mathbb{CN}(0, \sigma_h^2)$ and the diagonal element $R(i,i)$ is a scaled χ-square distributed random variable. (Refer [103].) More specifically, $2R(i,i)/\sigma_h^2$ is χ-square with $2(N-i+1)$ degrees of freedom. This means that it is the sum of squares of $2(N-i+1)$ i.i.d. standard real Gaussian random variables, i.e., variables having a $\mathcal{N}(0,1)$ distribution.

Therefore a lower right submatrix of $\begin{bmatrix} R \\ 0 \end{bmatrix}$ of size $(i+N-M)\times i$, say R_i, is statistically similar to it, i.e., it can be thought of as having arisen from the QR decomposition of an $(i+N-M)\times i$ matrix H_i having i.i.d. $\mathbb{CN}(0,\sigma_h^2)$ entries. Note that this is not to say that the H_i matrix is a submatrix of H. However, there exists H_i with the statistics mentioned above such that the QR decomposition of it gives us R_i, or, $H_i = Q_i \begin{bmatrix} R_i \\ 0_{N-M,i} \end{bmatrix}$ where Q_i is unitary of size $(i+N-M)\times(i+N-M)$. (For more on this, refer to [90]).

Recall z from Section 6.3. We have $z = Q^*x - \begin{bmatrix} R \\ 0 \end{bmatrix} s = \begin{bmatrix} R \\ 0 \end{bmatrix} (\tilde{s} - s) + Q^*v$. Define $w = Q^*v$. Clearly, w has the same statistics as v, i.e., i.i.d. $\mathbb{CN}(0,1)$ entries. Introduce $v_i = Q_i w^{i+N-M}$. Now v_i is of length $(i+N-M)$. (It is not necessarily a sub-vector of v.) As in the case of w, v_i will also have i.i.d. $\mathbb{CN}(0,1)$ entries. We can now write w^{i+N-M} as $Q_i^* v_i$.

Define $\gamma_i = \sum_{j=1}^{i+N-M} \lambda_j$ for $i = 1, \ldots, M$. Note that γ_i is the squared-norm of z^{i+N-M}. Also, we have s^i and $\tilde{s}^i$ as the lower length-i subvectors of s and $\tilde{s}$ respectively. From the above arguments, we have $z^{i+N-M} = \begin{bmatrix} R_i \\ 0_{N-M,i} \end{bmatrix} (\tilde{s}^i - s^i) +$

$Q_i^* v_i$. Therefore

$$\begin{aligned}\gamma_i = \|z^{i+N-M}\|^2 &= \left\| \begin{bmatrix} R_i \\ 0_{N-M,i} \end{bmatrix} (\tilde{s}^i - s^i) + Q_i^* v_i \right\|^2 \\ &= \left\| Q_i \begin{bmatrix} R_i \\ 0_{N-M,i} \end{bmatrix} (\tilde{s}^i - s^i) + Q_i Q_i^* v_i \right\|^2 = \left\| H_i(\tilde{s}^i - s^i) + v_i \right\|^2 .\end{aligned}$$

But it is clear that the vector $H_i(\tilde{s}^i - s^i) + v_i$ has i.i.d. $\mathbb{CN}(0, \sigma_v^2 + \sigma_h^2 \|s^i - \tilde{s}^i\|^2)$, i.e., $\mathbb{CN}(0, 1/c_i)$ entries. Therefore γ_i is a scaled χ-square distributed random variable. More specifically, $2c_i\gamma_i$ is χ-square with $2i$ degrees of freedom. This means that it is the sum of squares of $2i$ i.i.d. standard real Gaussian random variables, i.e., variables having a $\mathcal{N}(0,1)$ distribution. The expressions for the characteristic function of these are standard and we have $Ee^{j\alpha\gamma_i} = \frac{1}{(1-\frac{j\alpha}{c_i})^{i+N-M}}$.

For λ_i where $i > (N-M)$, note that $\gamma_{i-N+M} = \lambda_i + \gamma_{i-1-N+M}$. Moreover, since the λ_is are independent, so are λ_i and $\gamma_{i-1-N+M}$. Therefore $Ee^{j\alpha\gamma_{i-N+M}} = Ee^{j\alpha\lambda_i + j\alpha\gamma_{i-1-N+M}} = Ee^{j\alpha\lambda_i} Ee^{j\alpha\gamma_{i-1-N+M}}$. Thus

$$Ee^{j\alpha\lambda_i} = \frac{Ee^{j\alpha\gamma_{i-N+M}}}{Ee^{j\alpha\gamma_{i-1-N+M}}} = \frac{(1 - \frac{j\alpha}{c_{i-1-N+M}})^{i-1}}{(1 - \frac{j\alpha}{c_{i-N+M}})^{i}}. \tag{6.33}$$

For $i \leq (N-M)$ it is easy to see that λ_i is the squared norm of the the $(N-i+1)$th entry of Q^*v. Q^*v has the same statistics as v, i.e., i.i.d. entries, each with distribution $\mathbb{CN}(0,1)$. With this the characteristic function of λ_i is clearly $\frac{1}{1-\frac{j\alpha}{c_0}}$.

With this and Fourier inversion we get Table 6.1.

6.10.2 Derivation of Generating Function of Theorem 6.2

Theorem 6.4. *For $s, \tilde{s} \in \mathcal{S}^{k\times 1}$, the number of solutions to $\|s^k - \tilde{s}^k\|^2 = n$, averaged over all possible values of $\tilde{s}$ (as defined by $r_k^L(n)$ in (6.22)) is given by the coefficient of x^n in $(G_L(x))^k$ where $G_L(x) = \frac{1}{L^2}\left(L + \sum_{j=1}^{L-1} 2(L-j)x^{j^2}\right)^2$. Recall that $\mathcal{S} = \left\{a + jb \mid a, b \in \left\{-\frac{L-1}{2}, -\frac{L-3}{2}, \cdots, \frac{L-3}{2}, \frac{L-1}{2}\right\}\right\}$.*

Proof. For any complex vector x of length k define the vector x_{real} as a real vector of

length $2k$ where $x_{\text{real}}(2j-1) = \Re(x(j))$ and $x_{\text{real}}(2j) = \Im(x(j))$ for $j = 1, \ldots, k$.

Let $r = s^k - \tilde{s}^k$ where $s^k, \tilde{s}^k \in \mathcal{S}^{k\times 1}$. Then define r_{real} as above. Also define $\mathcal{S}_{\text{real}} = \left\{-\frac{L-1}{2}, -\frac{L-3}{2}, \cdots, \frac{L-3}{2}, \frac{L-1}{2}\right\}$.

Consider an arbitrary entry of r_{real}, say $r_{\text{real}}(j)$. For a fixed $\tilde{s}^k$, $\tilde{s}^k_{\text{real}}(j)$ is known. Say $\tilde{s}^k_{\text{real}}(j) = t \in \mathcal{S}_{\text{real}}$, then $r_{\text{real}}(j)$ takes all values in $\mathcal{S}_t = \mathcal{S}_{\text{real}} - t$. Define $q_t = \sum_{j\in\mathcal{S}_t} x^{j^2}\ \forall t \in \mathcal{S}_b$. Associate with a fixed vector $\tilde{s}^k$ the product $q(\tilde{s}^k) = \prod_{j=1}^{2k} q_{\tilde{s}^k_b(j)}$. Clearly, for this fixed $\tilde{s}^k$, the number of solutions to $\|s^k - \tilde{s}^k\|^2 = n$ is the coefficient of x^n in $q(\tilde{s}^k)$.

Since all the L^{2k} possible $\tilde{s}^k \in \mathcal{S}^{k\times 1}$ are assumed equally likely, the 'average' number of solutions to $\|s^k - \tilde{s}^k\|^2 = n$ is given by the coefficient of x^n in

$$\begin{aligned}
\frac{1}{L^{2k}} \sum_{\tilde{s}^k\in\mathcal{S}^{k\times 1}} q(\tilde{s}^k) &= \frac{1}{L^{2k}} \sum_{\tilde{s}^k\in\mathcal{S}^{k\times 1}} \prod_{j=1}^{2k} q_{\tilde{s}^k_{\text{real}}(j)} \\
&= \frac{1}{L^{2k}} \sum_{\sum_{t\in\mathcal{S}_{\text{real}}}\alpha_t = 2k} \binom{2k}{\{\alpha_t | t\in\mathcal{S}_{\text{real}}\}} \prod_{t\in\mathcal{S}_{\text{real}}} q_t^{\alpha_t} \\
&= \frac{1}{L^{2k}} \left(\sum_{t\in\mathcal{S}_{\text{real}}} q_t\right)^{2i} = \frac{1}{L^{2k}} \left(L + \sum_{j=1}^{L-1} 2(L-j)x^{j^2}\right)^{2i}
\end{aligned}$$

where $\binom{2k}{\{\alpha_t | t\in\mathcal{S}_{\text{real}}\}}$ is the multinomial coefficient given by $\binom{2i}{\alpha_{\frac{-(L-1)}{2}}, \alpha_{\frac{-(L-3)}{2}}, \cdots, \alpha_{\frac{L-3}{2}}, \alpha_{\frac{L-1}{2}}}$. Finally, we define $G_L(x) = \frac{1}{L^2}\left(L + \sum_{j=1}^{L-1} 2(L-j)x^{j^2}\right)^2$. □

We note here that this is closely related to the problem of representing integers as a sum of squares. For more on this refer to [90].

6.10.3 Derivations for Section 6.7.3

6.10.3.1 Proof of (6.29) and (6.30)

Let $I(\Delta) = \int_0^\Delta f(x)g(x)dx$. Let $U(x) = \int_0^x f(t)dt$. Let $h(U(x)) = x$. Then, changing variables in $I(\Delta)$, we have

$$I(\Delta) = \int_0^{U(\Delta)} g(h(U))dU = G(U(\Delta)) - G(0)$$

where $\frac{dG(x)}{dx} = g(h(x))$. Consider Taylor expansions for $G(U(\Delta))$ and $G(0)$ around $\frac{1}{2}U(\Delta)$. We have

$$\begin{aligned}
& I(\Delta) \\
= \;& G(U(\Delta)) - G(0) \\
= \;& G\left(\frac{1}{2}U(\Delta)\right) + \frac{1}{2}U(\Delta)\frac{dG(U(x))}{dU(x)}\bigg|_{U(x)=\frac{1}{2}U(\Delta)} + \left(\frac{1}{2}U(\Delta)\right)^2 \frac{1}{2}\frac{d^2G(U(x))}{(dU(x))^2}\bigg|_{U(x)=\frac{1}{2}U(\Delta)} + O([U(\Delta)]^3) \\
& - \left(G\left(\frac{1}{2}U(\Delta)\right) - \frac{1}{2}U(\Delta)\frac{dG(U(x))}{dU(x)}\bigg|_{U(x)=\frac{1}{2}U(\Delta)} + \left(\frac{1}{2}U(\Delta)\right)^2 \frac{1}{2}\frac{d^2G(U(x))}{(dU(x))^2}\bigg|_{U(x)=\frac{1}{2}U(\Delta)} + O([U(\Delta)]^3)\right) \\
= \;& U(\Delta)\frac{dG(U(x))}{dU(x)}\bigg|_{U(x)=\frac{1}{2}U(\Delta)} + O([U(\Delta)]^3) \\
= \;& g(h(U(x')))\int_0^\Delta f(x)dx + O\left(\left[\int_0^\Delta f(x)dx\right]^3\right) \\
= \;& g(x')\int_0^\Delta f(x)dx + O\left(\left[\int_0^\Delta f(x)dx\right]^3\right)
\end{aligned}$$

where x' is obtained by solving $U(x') = \frac{1}{2}U(\Delta)$ or $\int_0^{x'} f(t)dt = \frac{1}{2}\int_0^\Delta f(t)dt$. Note that the error term above is cubic in $\int_0^\Delta f(x)dx$. By a similar Taylor expansion around any $x'' \in [0, \Delta]$, we have $I(\Delta) = g(x'')\int_0^\Delta f(x)dx + O\left(\left[\int_0^\Delta f(x)dx\right]^2\right)$. Here, the error term is quadratic in $\int_0^\Delta f(x)dx$ rather than cubic, which is worse than before. In any case, for $\int_0^\Delta f(x)dx \ll 1$ we have

$$I(\Delta) \approx g(x')\int_0^\Delta f(x)dx. \tag{6.34}$$

6.10.3.2 Proof of (6.27) and (6.28)

Recall $\beta_{i,j} = \sum_{k=i}^{j} \lambda_{k+N-M}$ for $1 \leq i \leq j \leq M$ where λ_i is as defined in Section (6.3).

$$P(s^k \in \mathcal{D}_k) = P(\beta_{1,1} \leq r_1^2, \ldots, \beta_{1,k} \leq r_k^2) = \int_0^\infty P(\beta_{1,1} \leq r_1^2, \ldots, \beta_{1,k} \leq r_k^2 | \beta_{1,1}) p_{\beta_{1,1}}(\beta_{1,1}) d\beta_{1,1} \tag{6.35}$$

Note that $\beta_{i,i} \rightarrow \beta_{i,i+1} \rightarrow \ldots \rightarrow \beta_{i,j}$ with $1 \leq i \leq j \leq M$ is a Markov chain. Hence

$$P(s^k \in \mathcal{D}_k) = \int_0^\infty P(\beta_{1,1} \leq r_1^2 | \beta_{1,1}) p_{\beta_{1,1}}(\beta_{1,1}) \cdot P(\beta_{1,2} \leq r_1^2, \ldots, \beta_{1,k} \leq r_k^2 | \beta_{1,1}) d\beta_{1,1}.$$

With $P(\beta_{1,1} \leq r_1^2 | \beta_{1,1}) p_{\beta_{1,1}}(\beta_{1,1})$ as $f(\beta_{1,1})$ and $P(\beta_{1,2} \leq r_1^2, \ldots, \beta_{1,k} \leq r_k^2 | \beta_{1,1})$ as $g(\beta_{1,1})$, we use (6.34) to get

$$\begin{aligned} P(s^k \in \mathcal{D}_{IR,k}) &\approx P(\beta_{1,1} \leq r_1^2) P(\beta_{1,2} \leq r_2^2, \ldots, \beta_{1,k} \leq r_k^2 | \beta_{1,1} = \beta_{1,1}^{'}) \\ &= P(\beta_{1,1} \leq r_1^2) P(\beta_{2,2} \leq r_2^2 - \beta_{1,1}^{'}, \ldots, \beta_{2,k} \leq r_k^2 - \beta_{1,1}^{'}) \end{aligned}$$

where the optimum $\beta_{1,1}^{'}$ is obtained by solving $\int_0^{\beta_{1,1}^{'}} p_{\beta_{1,1}}(\beta_{1,1}) d\beta_{1,1} = \frac{1}{2} \int_0^{r_1^2} p_{\beta_{1,1}}(\beta_{1,1}) d\beta_{1,1}$ and the error is $O\left(\left[P(\beta_{2,2} \leq r_2^2 - \beta_{1,1}^{'}, \ldots, \beta_{2,k} \leq r_k^2 - \beta_{1,1}^{'}) \right]^3 \right)$.

Observe that the second term on the RHS is of the same form as the expression for $P(s^k \in \mathcal{D})$ in (6.35) and can be similarly approximated. In order to formalize this recursion, define $W_i = P(\beta_{i,j} \leq r_j^2 - \sum_{l=1}^{i-1} \beta_{l,l}^{'}$ for $j = i, \ldots, k)$ and $q_i = P(\beta_{i,i} \leq r_i^2 - \sum_{l=1}^{i-1} \beta_{l,l}^{'})$ where the $\beta_{i,i}^{'}$s are obtained successively by solving

$$\int_0^{\beta_{i,i}^{'}} p_{\beta_{i,i}}(\beta_{i,i}) d\beta_{i,i} = \frac{1}{2} \int_0^{r_i^2 - \sum_{l=1}^{i-1} \beta_{l,l}^{'}} p_{\beta_{i,i}}(\beta_{i,i}) d\beta_{i,i}. \tag{6.36}$$

The general recursion we have is $W_{t-1} = q_{t-1} W_t + O(W_t^3) \quad \text{for} \quad t = 2, \ldots, k$ where we start with $W_1 = q_1 W_2 + O(W_2^3)$, then use the recursion for W_2 and so on. Ignoring lower-order terms we get $W_1 = q_1 \cdot q_2 \cdots q_k (1 + O(\frac{q_k^2}{q_{k-1}}))$, since $W_k = q_k$. Note that all the q_is are probabilities and hence small numbers. Therefore $P(s^k \in \mathcal{D}_k) = W_1 \approx q_1 \cdot q_2 \cdots q_k$. Note that if we do not solve (6.36) exactly, this approximation still holds, but we have $W_1 = q_1 \cdot q_2 \cdots q_k (1 + O(\frac{q_k}{q_{k-1}}))$.

To summarize:

$$P(s^k \in \mathcal{D}_k) = \prod_{i=1}^{k} P\left(\beta_{i,i} \leq r_i^2 - \sum_{l=1}^{i-1} \beta'_{l,l}\right)(1 + O(\frac{q_k^2}{q_{k-1}})) \tag{6.37}$$

where the $\beta'_{i,i}$s are as obtained in (6.36). Define $X_i = r_i^2 - \sum_{l=1}^{i-1} \beta'_{l,l}$. Note that $\beta_{i,i} = \lambda_{i+N-M}$. This gives (6.27).

function DECODE(x, H, r) — *x: received vector, H: known channel, r:vector of the radii schedule*

1. $H = Q \begin{bmatrix} R \\ 0 \end{bmatrix}$ — *QR decomposition of H. R has a real diagonal*
2. $t = Q^* x$, $y = [t_1, \ldots, t_M]$ — *y has the first M elements of t*
3. Initialize:$\mathcal{D} = \emptyset$, $y^{''} = r' = s = 0_{M\times 1}$ — *$\mathcal{D}$ as the set of vectors in the search region*
4. **while** $\mathcal{D} = \emptyset$ — *Repeat till search region is non-empty*
   $r =$GETNEWSCHEDULE — *Obtain new schedule with smaller ϵ*
   $\mathcal{D} =$DECREASE$(N, y, R, r, y^{''}, r', s, \mathcal{D})$ — *Call subroutine*
5. $s^* = \text{argmin}_{s\in\mathcal{D}} \|x - Hs\|^2$ — *Find closest element within search region*
6. **output** s^* — *Decoder output*

function DECREASE$(k, y, R, r, y^{''}, r', s, \mathcal{D})$ — *k: subdimension, s:vector under consideration, s^{M-k} is known. Subroutine finds possible values for s_k.*

1. **if** $k = 0$ — *Check if subdimension is zero*
   $\mathcal{D} = \mathcal{D} \cup \{s\}$ — *Conclude that s is inside the search region*
   return
2. **elseif** $k = N$ — *At the highest dimension*
   $r'_k = r_1, \quad y^{''}_k = y_1$ — *Initialize*
3. **else**
   $y^{''}_k = y_k - \sum_{j=k+1}^{N} r_{k,j} s_j$ — *Calculations to find permissible values of s_k*
   $r'_k = \left((r^2_{N-k+1} - r^2_{N-k}) + r'^2_{k+1} - (y^{''}_{k+1} - R_{k+1,k+1} s_{k+1})^2\right)^{1/2}$
4. $LB = \max\left(\left\lfloor \frac{r'_k + y^{''}_k}{r_{k,k}} - \frac{1}{2} \right\rfloor + \frac{1}{2}, -\frac{L-1}{2}\right)$,
   $UB = \min\left(\left\lceil \frac{-r'_k + y^{''}_k}{r_{k,k}} + \frac{1}{2} \right\rceil - \frac{1}{2}, \frac{L-1}{2}\right)$ — *Exact range of s_k with L-PAM*
5. **for** $n = LB : UB$ — *For each possible value of s_k*
   $s_k = n$ — *Assign that value to s_k*
   $\mathcal{D} =$DECREASE$(k-1, y, R, r, y^{''}, r', s, \mathcal{D})$ — *Call subroutine to find possible values of s_{k-1}*
6. **return**

Table 6.4: Pseudocode for the Increasing Radii Algorithm

δ	$M = 10$	$M = 20$	$M = 30$	$M = 40$	$M = 50$
$\epsilon = 0.1$	2.16	2.35	2.55	2.74	2.93
$\epsilon = 0.01$	4.09	4.29	4.48	4.67	4.96
$\epsilon = 0.001$	5.64	5.83	6.03	6.41	6.61
$\epsilon = 0.0001$	7.19	7.19	7.48	7.77	8.15

Table 6.5: Values of δ for various values of M and ϵ. For a pair of values M and ϵ, use the corresponding value of δ from the table and a schedule of $r_i^2 = (\delta \log M + i)\sigma_v^2$.

Chapter 7

Discussion

The problems and results presented in this thesis address wireless systems from a variety of perspectives. In this concluding chapter, we summarize our observations and point out directions for future work.

7.1 Models and Problem Formulation

The capacity region for a relay network, with one source, one destination, and one relay is unknown. This is because the most general relay network can have channels that depend on and interact with each other in very complicated ways. This tells us that even with a small network, the problem of capacity can be quite intractable. At the same time, the work of Kumar and Gupta [35] gives us great insight into the working of a network with, possibly, thousands of nodes. Similarly, the results of [4] give us precise capacity regions for a large class of networks, characterized by certain properties.

Both these results are possible because of simplifications made in either the modeling of the network, or the sort of questions that are asked about these networks. In the case of [35], as in the case of the ad hoc networks presented in Chapter 2 of this thesis, the models have been simplified to obtain a relatively homogeneous network. In these models, the connection strengths between nodes do not interact with each other in completely arbitrary manners. In the first case the interaction between them is based on geometric location and in the second case the connection strengths either

do not interact at all, or their interaction is based on distance. Also, the question that is addressed is less demanding than that of a precise capacity region – we only worry about the scaling behavior of the throughput, which is a quantity that is easier to deal with than the capacity while, at the same time, it has practical significance.

The key to the results of [4] and those of Chapter 4 is similar. In the former, the model is that of a wireline network with capacitated links, while in the latter the model incorporates broadcast and, possibly, interference, but is restricted to erasure links. In both cases, the simplicity of the models allows us to answer the question of precise capacity regions. Also, note that the networks considered here can have arbitrarily many nodes and topology.

Thus we see that modeling a network can make a critical difference in how amenable it can be to analysis. At the same time, models have to be related to physical networks and therefore cannot be simplified to the point where they have no bearing on real world scenarios. Interesting and relevant results are possible when a careful balancing act is done while modeling networks and asking questions about them.

7.2 Summary and Directions for Future Work

We now summarize the main results of this thesis and present directions for future work.

7.2.1 Ad Hoc Networks

In Chapter 2 we saw that introducing randomness in a network model can have beneficial results on the throughput. We proposed a model that deviated significantly from the distance-based model studied in the literature before us and in which connection strengths were drawn independently from a common distribution. The aggregate traffic flow turns out to be strongly dependent on the distribution that the connections are drawn from. We find that some distributions give encouraging scaling laws for the throughput. For instance, for n being the number of nodes in the network, the

throughput can scale as $\frac{n}{(\log n)^d}$ for some $d > 0$, which is only slightly sublinear and significantly better than the throughputs predicted by distance-based models.

In Chapter 3 we have also seen models that incorporate a mixture of distance-based and random effects and how these models allow us to move from the purely distance-based models, to the purely random models. More models that incorporate a combination of random and distance-based connections are also mentioned in Chapter 3. In particular, the three-scale and mixture models are proposed. Connections strengths for these are described, respectively, by

$$p_x(\gamma) = \begin{cases} f(\gamma) & \text{if } x \le r_1 \\ \frac{r_2 - x}{r_2 - r_1} f(\gamma) + \frac{x - r_1}{r_2 - r_1} \frac{\mu_\gamma r_2^m}{x^m} \exp(-\gamma \frac{x^m}{\mu_\gamma r_2^m}) & \text{if } r_1 < x \le r_2 \\ \frac{\mu_\gamma r_2^m}{x^m} \exp(-\gamma \frac{x^m}{\mu_\gamma r_2^m}) & \text{if } x > r_2 \end{cases}$$

and

$$p_x(\gamma) = \frac{R - x}{R} f(\gamma) + \frac{x}{R} \frac{1}{x^m} \exp(-\gamma x^m).$$

Throughput analysis for these models is an interesting extension of the current results.

Many questions remain to be answered in this general area. In Chapter 2 we have discussed upper bounds on the throughput achievable using a multihop strategy. Finding good upper bounds that are independent of the method of network operation is a challenge. The standard min-cut upper bounds are often too loose, therefore it is hard to say how much we may be sacrificing by sticking to a simple multihop strategy. Coming up with tight upper bounds as well as strategies for achieving them is an interesting line of research.

From a practical point of view, scheduling the relays that are used in communicating each of the messages should be done in a decentralized manner. Algorithms that allow nodes to decide for themselves which messages to relay and in which time slots are of great interest. Also, assuming that nodes start out only with the knowledge of their immediate connections, finding efficient ways for them to exchange the minimum required information in order to participate in network communication is

an important problem.

7.2.2 Wireless Erasure Networks and Network Coding

In Chapter 4, we presented a network with erasure links and incorporated the wireless features of broadcast and intereference. For these networks, and several multicast scenarios, we presented exact capacity regions. These had a nice max-flow, min-cut interpretation, and channel and network coding had to be done jointly to achieve this capacity region. In fact, some results of Chapter 5 show that separating the two can lead to a loss of rate. In addition, we showed that linear coding strategies, which lead to faster decoding, can also achieve the capacity region.

However, all our results hinge on the availability of side-information regarding the locations of erasure at the destination. What the capacity region would be in the absence of this side-information is an open problem. Also, generalizing our results to networks where the links are not erasure channels is of interest. One approach to take for these problems may be to ask what the appropriate side-information is that can give us a capacity result.

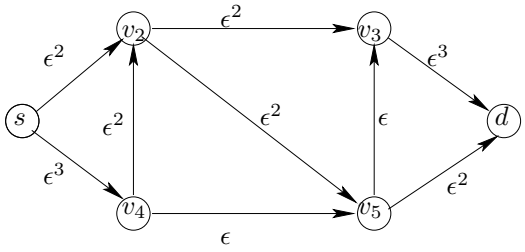
While many specific multicast problems have been solved, the most general case where multiple destinations want arbitrary subsets of the information available at multiple sources remains open, even in the case of wireline networks. This problem presents a challenging line of research.

7.2.3 Achieving Capacity with Simple Operations

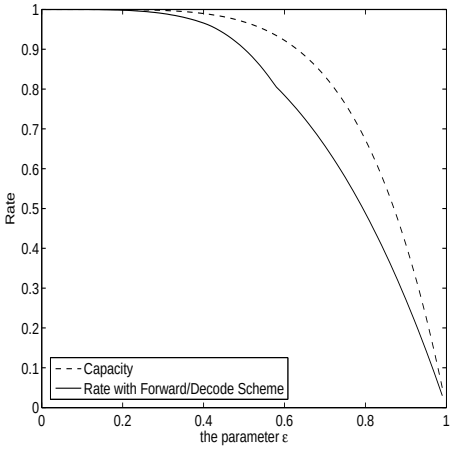
In Chapter 5, we considered erasure and Gaussian wireless networks with a single source and a single destination and began by showing that operating these by making each link or subnetwork error free is suboptimal. In other words, a loss of rate is incurred if we constrain every link or sub-network to be error free. Next we looked at a simple scheme in which nodes of these networks were allowed to either decode and then forward or simply forward the messages they receive and the goal was to maximize the rate from the source to the destination. The optimal operation for

each node was presented, and a decentralized algorithm that allowed each node to determine for itself this optimal operation was also proposed.

There is one difference between the erasure wireless network model of Chapter 4 and that of Chapter 5, in that the links in the latter can take erasures as inputs (and these are received as erasures) while the links of the former cannot. Ignoring this fact, we can compare the capacity of a network, as given by the theorems of Chapter 4, with the rates that can be achieved by the forward/decode schemes of Chapter 5. In Figure 7.2.3 we consider a simple network, with erasure links. The erasure probabilities for each channel are written in terms of a parameter ϵ that can vary from zero to one. The capacity of the network as well as the rates achieved by the forward/decode scheme are plotted.



(a) Capacity is $1 - \epsilon^5$



(b) Comparing network capacity and the rate of the optimal forward/decode scheme

Figure 7.1: The gap between the capacity and the rate achieved with the forward/decode scheme

We see that the practical scheme does very well when the erasure probabilities are small, but there is a gap between the rates that it can achieve and the capacity. Finding simple operations that can take us to capacity is an open problem. In general, given a set of simple, low complexity operations that each node is capable of performing, finding the optimal one for each node is an interesting question.

7.2.4 Decoding in Multiple Antenna Systems

In Chapter 6, we looked at the maximum-likelihood decoding problem for multiple antenna systems. This is an N-dimensional integer least squares problem in a probabilistic setting, where N is the number of (actual or virtual) antennas. Several heuristic methods of solving this problem using $O(N^3)$ operations are known, but the only known exact methods, such as the sphere decoder, have exponential complexity. We propose an algorithm that takes into account the statistical basis of the problem and prunes the search algorithm so as to reduce complexity. The pruning also makes the algorithm sub-optimal, and we characterize the loss of performance so as to be able to tradeoff complexity for performance. Our schemes can reduce complexity (with respect to a state-of-the-art sphere decoder) by a factor of 240 for a 50 antenna system with 4-QAM. With fewer antennas we expect smaller gains – a factor of 7 reduction for a 12-antenna system with 64-QAM. This is achieved while keeping performance within 0.1 dB of the optimal.

Although we have characterized the loss of performance with a particular pruning strategy, what the best pruning strategy is that allows for a certain tolerable loss of performance remains unclear. In other words, determining the optimum tradeoff between performance and complexity is an open problem. Another challenging question is that of finding the best pruning schedule for a given realization of the channel, rather than just basing it on the channel statistics.

Bibliography

[1] R. Gowaikar, A. F. Dana, R. Palanki, B. Hassibi, and M. Effros, "On the capacity of wireless erasure networks," *Proc. Intern. Symp. on Info. Theory*, 2004.

[2] A. F. Dana, R. Gowaikar, and B. Hassibi, "On the capacity region of broadcast over wireless erasure networks," *Proc. 42nd Annual Allerton Conf. on Communication, Control, and Computing*, Oct. 2004.

[3] E. C. van der Meulen, "Three-terminal communication channels," *Adv. Appl. Prob.*, vol. 3, 1971.

[4] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[5] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inform. Theory*, vol. 49, no. 2, pp. 371–381, 2003.

[6] N. Cai and R. W. Yeung, "Secure network coding," *Proc. of Intern. Symp. Inform. Theory*, 2002.

[7] T. M. Cover, "Broadcast channels," *IEEE Trans. Inform. Theory*, vol. 18, Jan. 1972.

[8] T. M. Cover and A. A. El Gamal, "Capacity theorems for the relay channel," *IEEE Trans. Inform. Theory*, vol. 25, pp. 572–584, Sep. 1979.

[9] L.-L. Xie and P. R. Kumar, "An achievable rate for multiple level relay channel," *IEEE Trans. Inform. Theory*, vol. 51, no. 4, pp. 1348-1358, April 2005.

[10] G. Kramer, M. Gastpar, and P. Gupta, "Cooperative strategies and capacity theorems for relay networks," *IEEE Trans. Inform. Theory*, Submitted Feb. 2004.

[11] Y. Wu, P. A. Chou, and S.-Y. Kung, "Minimum-energy multicast in mobile ad hoc networks using network coding," *submitted to IEEE Trans. on Communications*, Mar. 2004.

[12] D. S. Lun, M. Médard, T. Ho., and R. Koetter, "Network coding with a cost criterion," *Proc. Intern. Symp. on Info. Theory and its Appl.(ISITA 2004)*, Oct. 2004.

[13] M. Luby, "LT codes," *Proc. 43rd Annual IEEE Symp. on Found. of Computer Science*, Nov. 2002.

[14] D. S. Lun, M. Médard, and M. Effros, "On coding for reliable communication over packet networks," *Proc. 42nd Annual Allerton Conf. on Communication, Control, and Computing*, Oct. 2004.

[15] R. W. Yeung, *A First Course in Information Theory*, Kluwer Academic/Plenum Publishers, 2002.

[16] V I. Levenshtein, "Binary codes capable of correcting deletions, insertion and reversals," *Soviet Physics-Doklady*, vol. 10, Feb. 1966.

[17] J. D. Ullman, "On the capabilities of codes to correct synchronization errors," *IEEE Trans. Inform. Theory*, vol. 13, Jan. 1967.

[18] S. N. Diggavi and M. Grossglauser, "On transmission over deletion channels," *Proc. 39th Annual Allerton Conf. on Communication, Control, and Computing*, 2001.

[19] R. Gowaikar, A. F. Dana, B. Hassibi, and M. Effros, "Practical schemes for wireless networks operation," *submitted to IEEE Trans. on Communication*, 2004.

[20] A. F. Dana, R. Gowaikar, B. Hassibi, M. Effros, and M. Médard, "Should we break a wireless network into sub-networks?," *Proc. 41st Annual Allerton Conf. on Communication, Control, and Computing*, 2003.

[21] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: efficient protocols and outage behavior," *IEEE Trans. Inform. Theory*, Accepted for publication, Apr. 2004.

[22] D. Slepian and J. K. Wolf, "Noiseless coding for correlated information sources," *IEEE Trans. Inform. Theory*, Jul. 1973.

[23] A. D. Wyner and J. Ziv, "The rate-distortion function for source coding with side-information at the receiver," *IEEE Trans. Inform. Theory*, Jan. 1976.

[24] A. F. Dana and B. Hassibi, "The capacity region of multiple input erasure broadcast channel," *Proc. Intern. Symp. on Info. Theory*, Sep. 2005.

[25] F. Baccelli, M. Klein, M. Lebourges and S. Zuyev, "Stochastic geometry and architecture of communication networks," *J. Telecommunication Systems*, vol. 7, pp. 209–227, 1997.

[26] B. Bollobás, *Random Graphs*, 2nd ed., Cambridge: University Press, 2001.

[27] O. Dousse and P. Thiran, "Connectivity versus capacity in dense ad hoc networks," *Proc. 23rd INFOCOM*, Hong Kong, Mar. 2004.

[28] G. J. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas," *Bell Labs. Tech. J.*, vol. 1, no. 2, pp. 41–59, 1996.

[29] M. Gastpar and M. Vetterli, "On the capacity of wireless networks: the relay case," *Proc. 21st INFOCOM*, New York, Jun. 2002, pp. 1577–1586.

[30] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad-hoc wireless networks," *IEEE/ACM Trans. on Networking*, vol. 10, pp. 477–486, Aug. 2002.

[31] R. Hekmat and P. van Mieghem, "Degree distribution and hopcount in wireless ad-hoc networks," *Proc. 11th IEEE Int. Conf. on Networks (ICON)*, Sydney, Australia, pp. 603–609, Sep. 2003.

[32] R. Hekmat and P. van Mieghem, "Study of connectivity in wireless ad-hoc networks with an improved radio model," *Proc. 2nd Workshop on Model. and Optim. in Mobile, Ad Hoc and Wireless Networks*, Cambridge, UK, Mar. 2004.

[33] O. Léveque and E. Telatar, "Information theoretic upper bounds on the capacity of large extended ad hoc wireless networks," *to appear in the IEEE Trans. on Info. Theory.*

[34] M. Franceschetti, O. Dousse, D. Tse, and P. Thiran, "On the throughput capacity of random wireless networks," submitted to *IEEE Trans. Info. Theory*, 2004.

[35] P. Gupta and P. R. Kumar "The capacity of wireless networks," *IEEE Trans. Info. Theory*, vol. 46, pp. 388–404, Mar. 2000.

[36] P. Gupta and P. R. Kumar, "Towards an information theory of large networks: an achievable rate region," *IEEE Trans. Info. Theory*, vol. 49, pp. 1877–1894, Aug. 2003.

[37] I. E. Telatar, "Capacity of multi-antenna Gaussian channels," *Eur. Trans. Telecom.*, vol. 10, pp. 585–595, Nov. 1999.

[38] O. Tonguz and G. Ferrari, "Connectivity and transport capacity in ad hoc wireless networks," *Proc. 32nd IEEE Comm. Theory Workshop*, Mesa, AZ, Apr. 2003.

[39] L.-L. Xie and P. R. Kumar, "A network information theory for wireless communication: Scaling laws and optimal operation," *IEEE Trans. Info. Theory*, vol. 50, pp. 748–767, May 2004.

[40] S. Toumpis and A. Goldsmith, "Capacity bounds for large wireless networks under fading and node mobility," *Proc. 41st Allerton Conf. on Comm. Cont. and Comp.*, Monticello, pp. 1369–1378, Oct 2003.

[41] S. Weber, X. Yang, J. Andrews, and G. de Veciana, "Transmission capacity of wireless ad hoc networks with outage constraints," submitted to *IEEE Trans. on Info. Theory.*

[42] F. Chung and L. Lu, "The diameter of random sparse graphs," *Advances in Appl. Math*, vol. 26, pp. 257–279, 2001.

[43] A. Z. Broder, A. M Frieze, S. Suen and E. Upfal, "An efficient algorithm for the vertex-disjoint paths problem in random graphs," *Proc 7th Symp. Discrete Algorithms*, Atlanta, 1996, pp 261– 268.

[44] T. H. Cormen, C. E Leiserson, R. L. Rivest, and C. Stein *Introduction to Algorithms*, 2nd ed., The MIT Press, 2001.

[45] J. M. Abram and I. B. Rhodes, "Some shortest path algorithms with decentralized information and communication requirements," *IEEE Trans. Automatic Control*, vol. 27, pp. 570–582, 1982.

[46] Feng Xue, Liang-Liang Xie and P. R. Kumar, "The transport capacity of wireless networks over fading channels," *IEEE Trans. on Info. Theory,* vol. 51, no. 3, pp. 834–847, Mar 2005.

[47] A. Jovicic, P. Viswanath and S. R. Kulkarni, "Upper bounds to transport capacity of wireless networks," *IEEE Trans. Info. Theory,* vol. 50, no. 11, pp. 2555–2565, Nov. 2004.

[48] D. Miorandi and E. Altman, "Coverage and connectivity of ad hoc networks presence of channel randomness" *Proceedings of the IEEE INFOCOM* 2005, pp 491 – 502

[49] P. Balister, B. Bollobás, and M. Walters, "Continuum percolation with steps in an annulus," *Ann. Appl. Prob.,* vol. 14, no. 4, pp. 1869–1879, 2004

[50] M. Franceschetti, L. Booth, M. Cook, J. Bruck, and R. Meester, "Continuum percolation with unreliable and spread out connections" *Journal of Statistical Physics,* 118 (3/4), Feb. 2005.

[51] R. Gowaikar, B. Hochwald, and B. Hassibi, "An Achievability Result for Random Networks," *Proc. IEEE ISIT 2005*, Adelaide, Australia, pp 946-950.

[52] R. Gowaikar and B. Hassibi, "On the Achievable Throughput in Two-Scale Wireless Networks," *Proc. IEEE ISIT 2006*, Seattle, USA

[53] M. Penrose, *Random Geometric Graphs*, Oxford University Press, 2003.

[54] R. Gowaikar, B. Hochwald and B. Hassibi, "Communication over a wireless network with random connections," *IEEE Transactions on Information Theory*, July 2006.

[55] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain and L. Tolhuizen, "Polynomial time algorithms for multicast network code construction," *submitted to IEEE Transactions on Information Theory.*

[56] T. Ho, M. Médard, M. Effros, and R. Koetter, "Network coding for correlated sources," *Annual Conference on Information Sciences and Systems,* March 2004.

[57] T. Ho, B. Leong, R. Koetter, M. Médard and M. Effros, "Byzantine modification detection in multicast networks using randomized network Coding," *Proc. IEEE ISIT 2004.*

[58] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, pp. 782–795, Oct. 2003.

[59] B. Schein and R. Gallager, "The Gaussian parallel relay network," *Proc. IEEE ISIT 2000.*

[60] M. Gastpar and M. Vetterli, "On the capacity of wireless networks: the relay case," *IEEE INFOCOM*, vol. 3, pp. 1577–1586, June 2002.

[61] R. Koetter and M. Médard, "Beyound routing: an algebraic approach to network coding," *Proceedings of INFOCOM*, vol. 1, pp. 122–130, 2002

[62] A. F. Dana, M. Sharif, B. Hassibi, and M. Effros, "Is broadcast plus multi-access optimal for Gaussian wireless networks with fading?," *37th Asilomar Conference on Signals, Systems and Computers,* 2003.

[63] A. F. Dana, R. Gowaikar, R. Palanki, B. Hassibi, and M. Effros, "Capacity of erasure wireless networks," *IEEE Transactions on Information Theory*, vol. 52, pp. 789–804, Mar 2006.

[64] S.-Y. R. Li, R. W. Yeung and N. Cai, "Linear network coding," *IEEE Trans. Info. Theory*, vol. 49, pp. 371–381, Feb. 2003.

[65] M. Effros, M. Médard, T. Ho, S. Ray, D. Karger, and R. Koetter, "Linear network codes: a unified framework for source channel, and network coding," *invited paper to the DIMACS workshop on Network Information Theory*, 2003.

[66] L. R. Ford, Jr., and D. R. Fulkerson, *Flows in Networks,* Princeton University Press, Princeton, NJ, 1962.

[67] T. Ho, M. Médard, and R. Koetter, "An information theoretic view of network management," *submitted to IEEE Transactions Information Theory*

[68] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46/4, pp. 1204–1216, 2000.

[69] G. Caire and D. Tuninetti, "The throughput of hybrid-ARQ protocols for the Gaussian collision channel," *IEEE Transactions on Information Theory*, vol. 47/5, pp. 1971–1988, July 2001.

[70] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity. Part I System description. Part II Implementation aspects and performance analysis," *IEEE Transactions on Communications*, vol. 51, pp. 1927–1948, Nov. 2003.

[71] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley, New York, 1991.

[72] J. H. van Lint and R. M. Wilson, *A Course in Combinatorics*, Cambridge University Press, 2001.

[73] D. B. West, *Introduction to Graph Theory*, Prentice Hall, 1996.

[74] T. Ho, R. Koetter, M. Médard, D. Karger, and M. Effros, "The benefits of coding over routing in randomized setting," *Proc. IEEE ISIT*, 2003.

[75] H. L. Bodlaender and T. Wolle, "A Note on the Complexity of Network Reliability Problems," Download available at
`http://www.cs.uu.nl/research/techreps/aut/thomasw.html`

[76] L. G. Valiant, "The complexity of enumeration and reliability problems," *SIAM Journal of Computing*, vol. 8, pp. 410–421, 1979.

[77] J. S. Provan and M. O. Ball, "The complexity of counting cuts and of computing the probability that a graph is connected," *SIAM Journal of Computing*, vol. 12/4, pp. 384–393, 1983.

[78] G. J. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas," *Bell Labs. Tech. J.*, vol. 1, no. 2, pp.41–59, 1996.

[79] B. Hassibi and B. Hochwald, "High-rate codes that are linear in space and time," *IEEE Trans. Info. Theory*, vol. 48, no. 7, pp. 1804-1824, July 2002.

[80] M. O. Damen, A. Chkeif and J.-C. Belfiore, "Lattice code decoder for space-time codes," *IEEE Comm. Let.*, pp. 161–163, May 2000.

[81] B. Hassibi, "An efficient square-root algorithm for BLAST," *submitted to IEEE Trans. Sig. Proc.*, 2000 Download available at `http://mars.bell-labs.com`.

[82] R. Kannan, "Improved algorithms on integer programming and related lattice problems," *Proc. 15th Annu. ACM Symp. on Theory of Computing*, pp. 193–206, 1983.

[83] J.C. Lagarias, H.W. Lenstra, and C.P. Schnorr, "Korkin-Zolotarev bases and successive minima of a lattice and its reciprocal," *Combinatorica*, vol. 10, pp. 333–348, 1990.

[84] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Mathematics of Computation*, vol. 44, no. 170, pp. 463–471, April 1985.

[85] J. Gross and J. Yellen, *Graph Theory and its Applications*, CRC Press, New York, 1998.

[86] M. Stojnic, H. Vikalo, and B Hassibi, "A branch and bound approach to speed up the sphere decoder," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. iii/429–iii/432, 2005

[87] A.H. Banihashemi and A.K. Khandani, "On the complexity of decoding lattices using the Korkin-Zolotarev reduced basis," *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 162–171, 1998.

[88] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Transactions on Information Theory*, vol. 48, no. 8, pp. 2201–2214, 2002.

[89] M. Ajtai, "Generating hard instances of lattice problems," *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pp. 99–108, 1996.

[90] H. Vikalo and B. Hassibi, "On the sphere decoding algorithm. I. Expected complexity II. Generalizations, second-order statistics, and Applications to Communications," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2806–2834, 2005.

[91] J. Jalden and B. Ottersten, "On the complexity of sphere decoding in digital communications," *IEEE Transactions on Signal Processing*, vol. 53, no. 4, pp. 1474–1484, 2005.

[92] C. Windpassinger, R. F. H. Fischer, and J. B. Huber, "Lattice-reduction-aided broadcast precoding," *IEEE Transactions on Communications*, vol. 52, no. 12, pp.2057–2060, 2004.

[93] K. Su and I. J. Wassell, "A New Ordering for Efficient Sphere Decoding," *IEEE International Conference on Communications*, pp. 1906–1910, 2005.

[94] W. Xu, Y. Wang, Z. Zhou, and J. Wang, "Joint ML channel estimation and data detection for STBC via novel sphere decoding algorithms," *IEEE Vehicular Technology Conference*, pp. 434–437, 2005

[95] W. Zhao and G. B. Giannakis, "Sphere decoding algorithms with improved radius search," *IEEE Transactions on Communications*, vol. 53, no. 7, pp. 1104–1109, 2005.

[96] R. Gowaikar and B. Hassibi, "Efficient near-ML decoding via statistical pruning," *IEEE International Symposium on Information Theory*, pp. 274, 2003.

[97] R. Gowaikar and B Hassibi, "Efficient statistical pruning for maximum likelihood decoding," *IEEE ICASSP*, vol. 5, pp. 49–52, 2003.

[98] R. Gowaikar and B. Hassibi, "Efficient maximum-likelihood decoding via statistical pruning," submitted to *IEEE Transactions on Signal Processing*, accepted.

[99] B. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Transactions on Communications*, vol. 51, no. 3, pp. 389–399, 2003.

[100] M. Grotschel, L. Lovasz, and A. Schriver, *Geometric Algorithms and Combinatorial Optimization*, Springer Verlag, 2nd ed., 1993.

[101] J. -Y. Cai, "On the average-case hardness of CVP," *IEEE Symposium on Foundations of Computer Science*, pp. 308–317, 2001.

[102] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner and H. Boelcskei, "VLSI implementation of MIMO detection using the sphere decoding algorithm," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 7. pp. 1566–1577, 2005.

[103] A. Edelman, *Eigenvalues and Condition Numbers of Random Matrices*, PhD Thesis, MIT, Dept. of Math., Boston, 1989.

[104] H. Vikalo and B. Hassibi, "On joint ML detection and decoding for linear block codes," *Proc. IEEE ISIT* pp. 275, Yokohama, Japan, 2003.