Throughput Scaling and Data Gathering in Wireless Networks

by

Awlok S. Josan

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering: Systems)
in The University of Michigan
2011

Doctoral Committee:

Professor David L. Neuhoff, Chair
Associate Professor Anna C. Gilbert
Associate Professor Mingyan Liu
Associate Professor Sandeep P. Sadanandarao

"This one is tricky. You have to use imaginary numbers like eleventeen and thirty-twelve."

Bill Waterson

Calvin and Hobbes

To my parents

ACKNOWLEDGEMENTS

There are a lot of people whose love, support and guidance have been invaluable to me during my time at Michigan. First, I would like to thank my advisor, Dave Neuhoff. His guidance and patience during all my years in graduate school has been amazing. He was available to meet anytime I wanted and read every piece of writing I sent his way. This dissertation would not have been possible without him. I also want to thank the other members of my committee. Sandeep Pradhan, whose course on information theory helped me conceptualizing my research. The sensor networks group with Mingyan Liu was highly influential in my understanding of ad hoc networks. Anna Gilbert, who always had a different take on the problem and always helped me with new ideas about my research.

A huge thanks goes to my family for their incredible support during all my years of graduate school. This dissertation is dedicated to my parents: Arjan Josan and Narinder Josan, for their love, support and encouragement. I would also like to thank my sister, Ramnik, for her support, especially during the first couple of years in the US.

Last, but most certainly not the least, a huge thanks to all my friends at Michigan. Going to tailgates and football games, all those coffee breaks and time spent at the Den was certainly some of the best in graduate school. I would run out of space if I try to thank everyone here, but I do want to thank a few people personally. Rahul, Kim, Emine, Tzeno, Sid - you guys have been here with me since the beginning and it has been a fun ride. Laura, you have been incredibly supportive during the entire process of writing. I would also like to thank Rajit, Danese, Cristina, Christine, Chris, Sean, Nancy, Aranu, Matt, Scott and Carmelo. I will treasure these memories and friendships for a long time.

Contents

List of Figures

ABSTRACT

Throughput Scaling and Data Gathering in Wireless Networks

by

Awlok S. Josan

Chair: David L. Neuhoff

In this dissertation we investigate three problems associated with wireless networks. First, we examine throughput scaling in random access communication networks. We consider extended networks, that is, networks in which the number of nodes and the area increases such that the density of nodes remains constant. Franceschetti et al. have recently shown that per-node throughput in a geographically expanding, ad hoc wireless network with $\Theta(n)$ randomly distributed nodes and multihop routing can be increased to $\Omega(\frac{1}{\sqrt{n}})$ from the $\Omega(\frac{1}{\sqrt{n \ln n}})$ scaling demonstrated in the seminal paper of Gupta and Kumar. In this dissertation we explore the dependence of this interesting result on the new features it introduced: (1) capacity-based link transmission bit-rates, rather than positive bit-rates when and only when signal-to-interference-and-noise ratio lies above a threshold; (2) hierarchical routing through communal highways, instead of separate routes for each source-destination pair; and (3) cell-based routes based on percolation rather than on straight lines. It is shown that throughput $\Omega(\frac{1}{\sqrt{n}})$ can be attained with a system, without highways, that uses percolation to establish, for each source-destination pair, a set of routes within a narrow routing corridor. It is also shown that throughput $\Omega(\frac{1}{\sqrt{n}})$ can be attained with the threshold link-rate model, provided that transmission powers are permitted to grow with n. The conclusions are that the improved throughput scaling is principally due to the percolation-based routing, which enables shorter hops and, consequently, less interference, and that the benefit of the capacity link-rate model is simply to permit the power to remain bounded, even as the network expands.

The second problem we examine is that of the reliability efficiency trade-off in wireless sensor networks. In these networks, in comparison to independent coding, distributed lossless coding of correlated sources offers the potential for sizable reductions in coding rates. However, since the decoder performs joint decoding of all sources, it is highly sensitive to encoder failures, as the loss of even one encoder may result in failure to decode all sources. To increase reliability in case of encoder failures, this dissertation considers distributed coding schemes such that decoding of one source depends only on encoded data from a subset of other sources. While the reliability efficiency trade-off was introduced and studied in earlier work, in this dissertation the problem is cast in a rigorous formulation. We also introduce a new class of schemes, which we call flexible, in comparison to previous schemes which are called rigid. The efficiency of these schemes is measured in terms of average rate, and the reliability in terms of *loss factor*, which is the expected fraction of sources lost at the decoder. In order to evaluate the average rates of these schemes, the Slepian-Wolf lossless source coding theorem is extended to the case where encoders may fail. Several new flexible coding schemes are introduced and the trade-off between the reliability and efficiency of these schemes is analyzed. It is also found that flexible schemes generally outperform rigid.

The final problem investigated is sensor placement and real time data gathering in wireless sensor networks. Here, each sensor observes samples of an underlying random process at a given location and communicates the observations to the collector. The collector then estimates the process over the entire network region. One question concerns the optimal placement of N sensors in order to minimize the mean squared error in reconstruction. While sampling has been extensively studied and many questions have been answered, this question has not. It is shown that for a one-dimensional Markov process with exponential autocorrelation, uniform placement of sensors is optimal. Next, we propose a simple algorithm for collection of data in a one-dimensional network. In this case, the underlying field is modeled as a spatio-temporal random process. In order to provide good reconstruction of the field, we want data from a large number of sensors, to get good spatial granularity, and small delay, so that the data is still useful in estimating the current field. However, since each sensor competes for time-slots to transmit to the collector, placing a large number of sensors, while improving spatial coverage, results in large delay for much of data. On the other hand, placing a smaller number of sensors while reducing delay, results in poor spatial coverage. Thus, there exists a trade-off between the number of sensors and the delay suffered by data. This suggests that there exists an optimal density of nodes that minimizes the error in reconstruction. We show that for a stationary process with a separable correlation model that is exponential in both spatial and temporal domain, the optimal density of sensors over a fixed area increases with increasing temporal correlation and decreases with increasing spatial correlation.

CHAPTER I

Introduction

1.1 Motivation

Wireless networks have become a major part of our daily lives. Examples of wireless networks include cellular phone networks, wi-fi, sensor networks, etc. Some of these networks have infrastructure, such as cellphone towers, that form a backbone and all data is routed through the backbone. In this scenario, if two users want to communicate, data travels from the first user to the cellphone tower through a wireless channel, then between two towers, either through a wireless channel or through a wired backbone, and finally to the second user through another wireless channel. These networks typically have powerful towers with large computing power and big antennas. The end users on the other hand, have comparatively smaller computing power and smaller antennas. On the other extreme are *wireless ad hoc networks*, in which users self-organize to form a connected network. In this kind of network, users communicate with each other using others to relay their data. Hence, in these networks all data communication is peer-to-peer. In typical wireless ad hoc networks, the difference in computing and communication capabilities of different nodes is much smaller compared to networks with infrastructure. In the middle of these extremes are *hybrid networks.* These networks have some limited infrastructure in the form of powerful nodes or wired backbone. However, in these networks data communication happens both peer-to-peer between the less powerful nodes and also between the users and the backbone. Networks in which every source communicates to the same common destination are called *many-to-one*, while those in which every node can act as a destination are called *one-to-one*. IN a similar fashion, there also exist networks in which each node communicates to a multitude of other nodes. These networks are generally called *one-to-many* or *many-to-many* depending on how many nodes are transmitting. In this dissertation, we focus on one-to-one and many-to-one networks. The final classification that we consider in this dissertation is whether the networks are structured to perform some specific task at a central location or whether they are structured to provide communication between any two nodes. Sensor networks, which are usually constructed to measure some underlying physical phenomenon, fall under the former category. On the other hand, cellular networks are constructed to provide communication capabilities between any two users.

In this dissertation, we will concentrate on two specific kinds of wireless networks: random access wireless ad hoc networks, as in Figure 1.1, and field gathering wireless sensor networks, as in 1.2. The "random" term in the random access wireless ad hoc networks refers to the choice of destination for each node. In these networks each node acts as a source of data. The node may want to communicate to any one of the other nodes in the network. Thus, the number of destinations in the network is on the same order as the number of users. Field gathering wireless sensor networks, which fall under the many-to-one wireless network classification, are structured to perform a specific task. Typically, these networks are constructed to measure some physical phenomenon, such as temperature, over a geographical region. These networks consists of a fixed number of nodes, each sensing the underlying phenomenon at its location. Each node then communicates is measurements to a central location, called the *collector*, either directly or by using other nodes as relays. Thus, each node in one of these networks has the same destination.

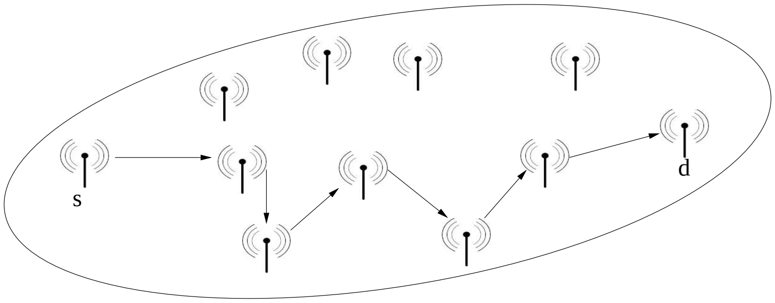


Figure 1.1: Random Access Wireless Ad-hoc Network. Source s communicates its data to the destination d using other nodes as relays.

1.2 Random Access Wireless Ad Hoc Networks

A random access wireless ad hoc network consists of a fixed number of users/nodes, spread over some geographical area. These nodes may be static or mobile. Each node in the network randomly chooses a destination to which it wants to communicate data. Communication between each source-destination is done in a multihop fashion using other nodes as relays.

Since these networks do not require an infrastructural support they can be used for battlefield deployment [1]. Another potential application is automated highway systems [2] in which cars communicate with each other to maintain safe distance from each other. These networks also have potential uses in disaster recovery where response teams need to communicate with each other [3]. These networks also have uses where a number of robots cooperatively work on some given task.

1.2.1 Challenges

In this section we study some of the challenges faced in designing random access wireless ad hoc networks.

Since random access wireless communication networks do not have any central control, there are a variety of problems that arise when designing and analyzing these networks. Some of the typical issues that designers face when constructing these networks are:

- Connectivity: Since these networks are self-organizing, an important issue that arises is whether each node can communicate with all other nodes, either directly or by using some other nodes as relays.

- Routing: In order for two nodes to communicate with each other, there needs to be a route, that is, a sequence of other nodes that can act as relays between the two nodes.

- Transmission Power: In order for a node to communicate with another node, it needs to transmit at high enough power. However, if all nodes transmit at high power there will be a lot of interference at the receiver. So the choice of transmission power is important in providing connectivity in the network.

- Scheduling: As stated before, in order to ensure successful transmissions from a transmitter to its intended receiver, we need to limit interference at the receiver from all other simultaneous transmissions. This means that all nodes cannot transmit simultaneously. Thus, there need to be strategies that allow nodes to decide when to transmit so that all of them, or at least a large percentage, are successful.

Two important metrics used to measure performance of random access networks are *throughput* and *delay*. Throughput is the number of bits per second that each node can transmit to its destination. Delay is the amount of time for the bits to travel from the source to the destination. In this dissertation, we focus on throughput and not delay. There are multiple factors that impact the attainable throughput of an ad hoc network. Some of the main ones are: the number of nodes, the position of the nodes, distances between nodes, the choices of sources and destinations, that is, which nodes are communicating with which others etc. As a result of this dependence on a wide variety of factors, computing the exact number of bits that a source can send to its destination is extremely hard.

1.2.2 Throughput Scaling

As a first step to finding the throughput of these networks, Gupta and Kumar [4] posed the *throughput scaling* problem. They investigated how the maximum attainable throughput of networks with a large number of users changed as the number of users in the network increased. The throughput of an ad hoc network is dependent on how many of the users can simultaneously transmit data to their receivers, how frequently they can transmit, and how many bits they try to transmit when scheduled to transmit. The data being transmitted by a user might be its own or it might be data that is being relayed for another user. Typically there will exist a trade-off between the number of users transmitting simultaneously and the number of bits each of them can transmit. The success of a transmission from a transmitter to a receiver depends on the interference from all other simultaneously transmitting nodes at a receiver. The lower the interference at a receiver, the better the chance of the transmission being successful. Thus, in order to improve the throughput of wireless ad hoc networks, we need to reduce the interference at each receiver. However as more users get added to the network, the interference within the network increases, as each new user wants to communicate data to its destination. There are multiple methods to reduce interference, two of which were employed by Gupta and Kumar. First, instead of communicating directly to its destination which may be far away, the source relays its data through a series of short hops using other nodes as relays. In this case, since transmissions need to take place over shorter distance, each transmitter can use a lower transmission power. This reduces interference at other receivers and allows more simultaneous transmissions to take place. Second, instead of all the nodes in the network simultaneously transmitting, there is a time division multiple access (tdma) sharing scheme, called a schedule, that allows only a limited set of transmitters to transmit in a given time-slot. This reduces both the interference at various receivers in the network, and also the number of simultaneous transmission that can take place. Designing the overall system requires finding routes from each source to its destination and setting up tdma schedules to ensure that each user in the network gets to successfully transmit at a satisfactory rate. In this dissertation, we give a constructive routing and scheduling strategy to analyze performance of large scale random access communication network.

1.2.3 Contributions: Analysis of Throughput Scaling

In Chapter II, we analyze the problem of throughput scaling in random access communication networks. These networks consist of a large number of users/sources, each wanting to communicate with a randomly chosen destination. We wish to find the maximum number of bits that each source can transmit to its destination. However, the random placement of users makes this an incredibly hard problem. As stated previously, Gupta and Kumar posed an interesting abstraction of this problem, in which they analyzed how the attainable throughput of these networks changed as the number of users grew. The answer to this question holds strong implications in designing a network. If the throughput is independent of the number of nodes in a network, it would mean that we could design large scale networks without suffering any ill effects in terms of decrease in throughput. Gupta and Kumar came up with a constructive scheme that showed that a scaling of $\Omega\left(\frac{1}{\sqrt{n \ln n}}\right)$ bits/slot was achievable. That is there exists a constant c such that for large enough n the throughput of the network is higher than $\frac{c}{\sqrt{n \ln n}}$. More recently, Franceschetti et al. [5] showed that this throughput scaling could be improved to $\Omega\left(\frac{1}{\sqrt{n}}\right)$. Along with some minor changes in the underlying network model, the three major changes that their work introduced were:

1. A capacity-based transmission rate model instead of a threshold-based model. Under their model, a transmitter could transmit to its intended receiver at rate equaling the Shannon capacity of the channel. Under the threshold-based model, transmissions were successful only if the receiver signal to interference and noise ratio (SINR) was above a fixed threshold.

2. A highway based routing scheme instead of a semi-straight line routing. In their routing scheme, they construct a number of horizontal and vertical highways. The data from each source is first relayed to a node on a horizontal highway. The data then travels first along the horizontal highway and then a vertical highway. From a node close to the destination on the vertical highway, it is finally relayed to the destination. This was in contrast to the routing scheme used by Gupta and Kumar which connected the source and destination by a straight line and used nodes withing a certain fixed distance of the line as relays.

3. The final significant difference between the models was the use of percolation to construct the above mentioned highways. These highways are a sequence of nodes, where each pair of nodes is called a hop. In this case, the data from from source to destination takes hops that are of significantly different length. In particular, the first hop from the source to a relay node on the horizontal highway (and the last hop from the relay node on a vertical highway to the destination) is a lot longer compared to hops on the highways. In contrast, in the scheme Gupta and Kumar proposed all the hop lengths were of a similar length. The hop lengths in their scheme were longer than the highway hops but shorter than the hop from source to relay node in Franceschetti.

In this dissertation, we construct a routing and scheduling scheme that achieves the throughput $\Omega(1/\sqrt{n})$ without a hierarchical highway structure and by using multiple routes for each source destination pair that lie in a narrow corridor. We also illustrate the impact of each one of the major changes described above. We show that the improvement in throughput is only due to the use of percolation to construct routes from each source to its destination. Since most hops on these paths have shorter lengths as compared to Gupta-Kumar, it allows more nodes in the network to simultaneously transmit, thereby improving throughput. We also show that the benefit of using the capacity-based model is that it allows the use of finite power at each node, that is, power does not increase as the size of the network increases. However, it requires the use of a more powerful receiver, as the receiver now needs to decode at extremely low SINRs. Finally, we show that there is no benefit achieved from using a highway based routing scheme instead of a semi-straight line routing.

1.3 Field Gathering Wireless Sensor Networks

Advances in hardware technology and wireless networking have made the possibility of low-cost, low-power and inexpensive wireless sensor networks (WSNs) a reality. For an overview of these networks see [6, 7]. Of particular interest are field gathering WSNs, Figure 1.2. These networks use a number of sensors deployed over a geographical area, called the *network region*, to measure some underlying natural phenomenon. The sensors measure the field at their location and then communicate their data to a central location, called the *collector*. The communication may done either directly by each sensor, or by making use of other sensors as relays. The collector, on receiving data from the nodes in the network, makes an estimate of the field in the entire network region. These networks have a vast array of applications, examples include:

- Habitat Monitoring [8]: Sensor networks can be used for habitat monitoring in order to study biodiversity and the bio-complexity. These networks can provide data at spatial granularity much better than that achieved using remote sensing

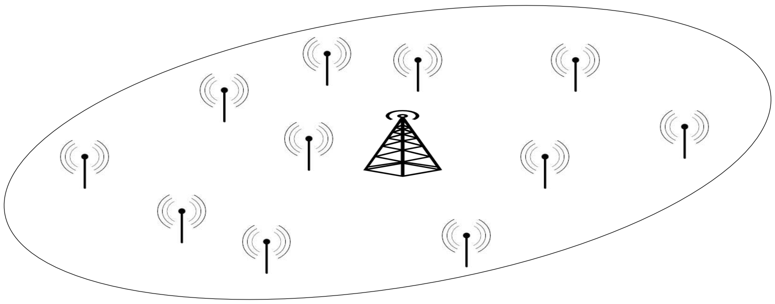


Figure 1.2: Many-to-one Wireless Sensor Networks. Each source communicates its data to the base-station.

satellites.

- Hazard Monitoring [9]: Self-organizing sensor networks can be deployed in hazard monitoring such as the current nuclear plant failure in Fukishima, Japan. These sensor spread in the vicinity of the plant can measure the radiation level and transmit to a central location.

- Soil Moisture [10]: Sensor networks are deployed to measure soil moisture. This data has potential uses in adaptive irrigation in agriculture or for calibration and validation of data gathered using remote sensing satellites.

- Parking [11, 12]: Sensors deployed into parking spots in a city can inform drivers the closest open parking spots.

- Battlefield Monitoring [13]: These networks in a battlefield can be used to detect and track enemy movement.

- Structure Health Monitoring [14]: Sensors embedded in civilian structure such as bridges can be used to measure beam-fatigue which would give early indications of failure.

For more applications of these networks see [15, 16, 17, 18, 19].

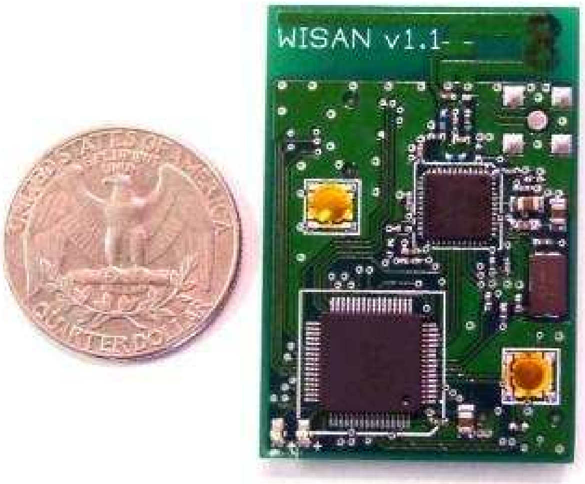


Figure 1.3: Typical size of a sensor node in comparison with a US quarter dollar. Image Courtesy Russ Nelson.

1.3.1 Challenges

Many of the challenges faced in designing WSNs arise from the fact that the sensor nodes have limited capabilities in terms of available power, transmission range, computing power etc. The limited capabilities are a direct consequence of their small size, see Figure 1.3. In order to understand the problems that we face in designing these networks, we first look at the different components of a sensor node and the functions that they perform.

A wireless sensor node, block diagram shown in Figure 1.4, typically consists of parts capable of performing the following functions:

- Sensing: This functionality in a node is performed by sensors and actuators. These produce a measurable response depending on the physical conditions that the node is trying to measure. Depending on the application these sensors can be active or passive. For example, in temperature monitoring the sensors would generally be passive, where as in battlefield detection a sensor might use active sonar or radar sensors.

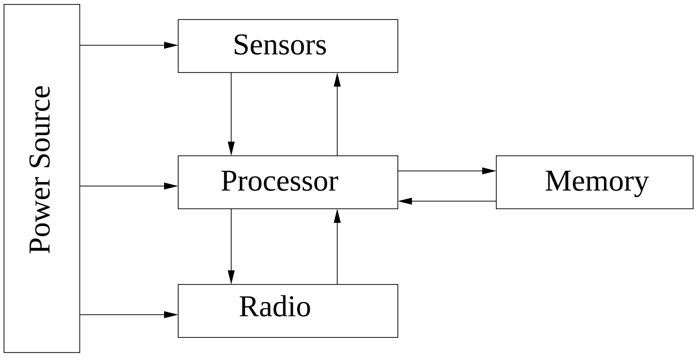


Figure 1.4: Typical components of a sensor node.

- Computing: The computing on a node usually consists of an ADC, a micro-controller and, optionally, a general purpose microprocessor. The ADC is used to convert the analog signal received from the sensor into a digital form, so that it be used in computations and can be transmitted easily. The micro-controller is used to control the sensor and the radio. For example, it determines when the radio should transmit and when to turn it off. The microprocessor may perform some computation on the sensed observation such as compressing the observations. These computations may be performed on the data obtained from its own sensors alone or in combination with data received from other nodes.

- Communication: Each node consists of a radio comprising of a transmitter and a receiver. The radio is used to communicate the data that originates at the node to the collector, either directly or by relaying it to another node. This radio is also used to receive and relay data for other nodes in the network.In most WSNs, communication of data is the most power hungry part of operation.

- Power: In order to perform all the above functions some amount of power is required. Since the nodes are not connected to a central location, this power has to come from an on board power source. Each node usually has a small battery to provide power. Some sensors also have an energy harvesting mechanism to recharge the battery.

- Storage: Nodes also have memory to provide data storage capabilities. The node may store its own past data or data received from other nodes in this memory.

1.3.2 Data Compression

One of the major limiting factors in the performance of wireless sensor networks is the amount of power available at each node. Power is required for all three of the operations that a typical sensor node performs: sensing, computation and communication. However, power required for the three operations is usually quite different. The biggest drain on power is usually the radio used for communication. The energy used for sensing, while depending on the type of sensor, is negligible for most passive sensors. The energy used in computation, while once again depending on the type of node, is quite small as compared to that used for communication. As an example, for some of the commercially available sensor nodes, "the energy spent in transmitting one bit of information is the same as that required by the microprocessor to execute approximately one thousand instructions [20, 21]", Croce et al [22]. Since replacing the batteries on these sensors in most applications would be physically hard and cost prohibitive, it is important to reduce the amount of data that each node needs to transmit.

There are multiple ways of reducing the amount of data that has to be transmitted to the collector, one of the obvious being data compression. The simplest method for compressing data in WSNs is where each node uses source coding to compress its own data without taking into consideration data at other nodes. Assuming that we have a fixed ADC at each node and the underlying probability distribution of the field is known, this can be done by using a coding scheme, such as scalar quantization followed by Huffman lossless coding [23, 24]. In case the source distribution is unknown universal source coding techniques [24, 25] may be employed. In many scenarios, the observations obtained at different nodes are also correlated. For example, if a sensor network is deployed to sense a temperature field, the readings obtained at sensors that are physically close together will be very similar. The correlation between observations at different nodes can also be exploited to compress data in multiple ways. As an example, consider the scenario, where node i communicates its observations to the collector using node j as a relay. Since node j has access to node i's data, it can conditionally encode [26] its own data given node i's observation. The collector upon receiving bits from both nodes, first decodes node i's data. Then using bits it received from node j and node i's data it decodes node j's data. Thus, in this case node j can reduce the number of bits it has to transmit to the collector.

Next, consider the scenario, where all nodes communicate their data directly to the collector without relaying. In this scenario, the observations made at one sensor are available only to the collector and not to any other sensor. In this case, since the collector has access to encoded data from all sensor nodes, the correlations between nodes can still be exploited by using *distributed coding* [24, 27]. In distributed coding, each encoder encodes its data independently. However, the decoder jointly decodes data from all nodes. In fact, assuming fixed ADCs at each node, the nodes can compress their data so that the sum of the number of bits communicated to the collector is the same as would have been in the case where every nodes data was available at every other node. In addition, distributed coding used with relaying to avoid decoding and re-encoding at each relay node.

1.3.3 Contributions: Reliability Efficiency Trade-off in Distributed Coding

As mentioned previously, transmission of data is one of the biggest drains on the battery of a sensor node and thus, we would want the sensors to transmit the smallest number of bits possible. In sensor networks where the observations at different sensors are correlated we can use distributed coding [27, 28] to reduce the rate at which each node transmits to the collector. In distributed coding, each node encodes its data independently. However, the collector jointly decodes data from all nodes. While the independent encoding of the data is extremely useful in WSNs, the joint decoding is an onerous requirement as it requires data from all nodes to be available at the decoder. In many practical WSNs data from some of the nodes may not reach the collector. This could happen, for instance, if the battery at a sensor node runs out of power, or if the channel conditions are such that a node is unable to transmit to the collector. Since loss of data from even one of the nodes could result in the decoder not being able to decode data for any of the nodes, the system is extremely vulnerable to node failures.

Marco and Neuhoff [29] introduced a framework for increasing robustness (reliability) of lossless distributed coding schemes at the expense of an increase in the coding rate, i.e. decreased efficiency. They measured the reliability of a scheme in terms of *loss factor* which is the number of encoders whose data cannot be reconstructed at the decoder, and the efficiency in terms of average rate. They propose a number of distributed coding schemes and analyzed the trade-off in average rate and loss factor. Of the schemes proposed, they found that Master-Slave offered performed the best in terms of average rate at a given loss factor.

In Chapter III of this dissertation, we cast the reliability-efficiency problem in a rigorous framework. In order to analyze distributed coding schemes in the presence of node failures, we extend the Slepian-Wolf lossless source coding theorem to the case where some of the encoders may fail. More significantly, we propose a new class of distributed coding schemes which we call *flexible*, as compared to the earlier schemes which can all classified as *rigid*. We view the decoder as being composed of a number of sub-decoders, one corresponding to each node. In rigid schemes, corresponding to each sub-decoder is a fixed subset of nodes. The sub-decoder can successfully decode only if encoded descriptions from all nodes in that subset are received. In flexible schemes, corresponding to a sub-decoder are multiple subsets of nodes. If encoded descriptions from all nodes in any one of these subsets are received, the sub-decoder can decode successfully. We analyze the new schemes in terms of average rate and robustness to node failures, i.e. loss factor. Finally, we provide numerical results for a one-dimensional network sensing a stationary Gauss-Markov process. It is seen that the flexible schemes generally outperform the rigid schemes by substantial amounts. Also, only the flexible schemes can achieve the lower bound on the loss factor for rates significantly smaller than the scheme where no distributed coding is used. Thus, flexible schemes can be highly efficient in terms of compressing data, while being robust to node failures.

1.3.4 Sensor Placement and Data Gathering

Another question in designing field gathering WSNs is where to place the sensors in the network region. Recall that the collector makes a reconstruction of the field in the entire network region, whereas observations from only those points where sensors are placed are available. This results in distortion in reconstruction. In the case of a given fidelity criteria, such as mean squared error (MSE), to measure distortion and a given fixed number of sensors, the question is where to place the sensors in order to optimize the reconstruction with respect to the given fidelity criteria. This optimal placement would generally be dependent on spatial correlation characteristics of the underlying field.

More generally, given a target MSE with which we want to reconstruct the field, the question becomes, how many nodes do we place and where should we place them. Also, in case the nodes are using lossy coding on the digitized data after the ADC, at what rate should each node encode its data. In general the more sensors we place, the lower the MSE. However, it might be the case that for certain placements the nodes need to transmit at a higher rate to achieve the target MSE.

Once the sensor positions are fixed, the question arises how to collect data from each of them. In many practical networks, all sensor will not be able to communicate there data to the collector simultaneously. This could be for a multitude of reasons, for example, in a slotted time system the collector may be able to receive data from only one sensor in each time slot. Since sensors now compete for time-slots to transmit data to the collector, placing a large number of sensors would cause much of the data to suffer a large delay. On the other hand placing a smaller number of sensors might not give good spatial coverage.

1.3.5 Contributions: Optimal One Dimensional Sensor Placement and Real Time Data Gathering

In Chapter IV we investigate the sensor placement and data gathering problem in wireless sensor networks. In this problem that we consider, we have a fixed number of sensors that we want to place over in a one-dimensional network of fixed area. We model the underlying field being measured as a stationary random process. Each sensor takes a measurement of this process at its location and communicates the observation to the collector. The collector, on receiving data from all the nodes, makes an estimate of the process in the entire network region. We use mean squared error, integrated over the entire network region, as our performance measure. We wish to find an optimal placement of the sensors, if it exists, where optimal means that it minimizes the mean squared error. This problem has very strong relation to sampling theory, and a lot of results in literature are available on similar problems. If the underlying field is modeled as a band-limited, signal the Shannon-Nyquist theorem [30, 31], also known as Whittaker-Kotelnikov-Shannon (WKS) theorem, says that the signal can be reconstructed from uniformly placed samples taken at a frequency that is twice the maximum frequency present in the signal. The theorem also has been extended to band-limited random processes by Balakrishnan [32]. However, the available literature for non-bandlimited random processes is very sparse. This elementary problem of placing a fixed number of sensors in a one-dimensional interval of fixed length has no known optimal solution. We show that for a one-dimensional Markov process with exponential correlation the optimal placement of sensors is uniform.

Having placed the sensors uniformly over the network region we propose a data gathering algorithm for real-time reconstruction of underlying process. For the real-time data gathering problem, we model the underlying field as a spatio-temporal random process. We assume a slotted-time system under which the each node can transmit to and receive data from its neighbors once during each time slot. The collector makes an estimate of the process over the entire network region in each time slot. We consider the simplest possible data gathering model for a one-dimensional network that captures the trade-off between the increasing number of sensors and delay. In our algorithm, each sensor linearly combines its own data to the data it is relaying for other sensors and relays the combined data to a neighboring sensor that is closer to the collector. We observe that under the constraint that each node can only communicate with its neighbor there exists an optimal density of nodes for the data gathering when the network region is one-dimensional. This density increases with increasing temporal correlation and decreases with increasing spatial correlation.

1.4 Dissertation Organization

As stated before, in this dissertation we examine three problems that afflict wireless networks. Chapter II examines the performance of random access wireless ad hoc networks, and Chapters III and IV investigate problems that are more specific to a subclass of wireless networks called field gathering wireless sensor networks. Specifically, Chapter III investigates the reliability efficiency trade-off, and Chapter IV explores the sensor placement and data gathering problems in wireless sensor networks.

Each of the above three chapters has its own introduction and list of references. Requisite notation is introduced in each of the chapters separately, too, thereby making each chapter self-contained.

References

[1] M. Gerla, "From battlefields to urban grids: New research challenges in ad hoc wireless networks," *Pervasive and Mobile Computing*, vol. 1, no. 1, pp. 77 – 93, 2005.

[2] P. Varaiya and S. Shladover, "Sketch of an ivhs systems architecture," in *Vehicle Navigation and Information Systems Conference*, vol. 2, 1991, pp. 909 – 922.

[3] G. Zussman and A. Segall, "Energy efficient routing in ad hoc disaster recovery networks," *Ad Hoc Networks*, vol. 1, no. 4, pp. 405 – 421, 2003.

[4] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. on Inform. Theory*, vol. 46, no. 2, pp. 388 – 404, Mar 2000.

[5] M. Franceschetti, O. Dousse, D. N. C. Tse, and P. Thiran, "Closing the gap in the capacity of wireless networks via percolation theory," *IEEE Trans. on Inform. Theory*, vol. 53, no. 3, pp. 1009 –1018, Mar 2007.

[6] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102 – 114, Aug. 2002.

[7] M. Tubaishat and S. Madria, "Sensor networks: an overview," *IEEE Potentials*, vol. 22, no. 2, pp. 20 – 23, Apr-May 2003.

[8] A. Cerpa, J. Elson, M. Hamilton, J. Zhao, D. Estrin, and L. Girod, "Habitat monitoring: application driver for wireless communications technology," in *Workshop on Data communication in Latin America and the Caribbean*, ser. SIGCOMM LA '01. New York, NY, USA: ACM, 2001, pp. 20–41. [Online]. Available: http://doi.acm.org/10.1145/371626.371720

[9] J. Kenney, D. Poole, G. Willden, B. Abbott, A. Morris, R. McGinnis, and D. Ferrill, "Precise positioning with wireless sensor nodes: Monitoring natural hazards in all terrains," in *IEEE International Conference on Systems, Man and Cybernetics, 2009.*, Oct. 2009, pp. 722 –727.

[10] M. Moghaddam, D. Entekhabi, Y. Goykhman, K. Li, M. Liu, A. Mahajan, A. Nayyar, D. Shuman, and D. Teneketzis, "A wireless soil moisture smart sensor web using physics-based optimal control: Concept and initial demonstrations," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 3, no. 4, pp. 522 –535, Dec. 2010.

[11] S. Lee, D. Yoon, and A. Ghosh, "Intelligent parking lot application using wireless sensor networks," in *International Symposium on Collaborative Technologies and Systems, 2008.*, May 2008, pp. 48 –57.

[12] http://sfpark.org/.

[13] T. Walker, M. Tummala, and J. McEachen, "Energy efficiency of centralized and distributed computation in unattended multi-hop wireless sensor networks for battlefield monitoring," in *2010 43rd Hawaii International Conference on System Sciences (HICSS),*, 2010, pp. 1 –9.

[14] K. Chintalapudi, T. Fu, J. Paek, N. Kothari, S. Rangwala, J. Caffrey, R. Govindan, E. Johnson, and S. Masri, "Monitoring civil structures with a wireless sensor network," *IEEE Internet Computing*, vol. 10, no. 2, pp. 26 – 34, Mar-Apr 2006.

[15] Y. Kawahara, N. Kawanishi, H. Morikawa, and T. Aoyama, "Top-down approach toward building ubiquitous sensor network applications," in *11th Asia-Pacific Software Engineering Conference, 2004*, Nov.-Dec. 2004, pp. 695 – 702.

[16] S. Kumar, A. Arora, and T. Lai, "On the lifetime analysis of always-on wireless sensor network applications," in *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference, 2005.*, nov. 2005, pp. 3 pp. –188.

[17] T. T. Hsieh, "Using sensor networks for highway and traffic applications," *IEEE Potentials*, vol. 23, no. 2, pp. 13 – 16, Apr-may 2004.

[18] J. Koskinen, P. Kilpelainen, J. Rehu, P. Tukeva, and M. Sallinen, "Wireless sensor networks for infrastructure and industrial monitoring applications," in *2010 International Conference on Information and Communication Technology Convergence (ICTC)*, Nov. 2010, pp. 250 –255.

[19] J. Nakazawa, J. Yura, T. Iwamoto, H. Yokoyama, and H. Tokuda, "Bridging sensor networks and the internet on cellular phones: Towards richer and easier-to-use sensor network applications for end-users," in *2009 Sixth International Conference on Networked Sensing Systems (INSS)*, Jun 2009, pp. 1 –4.

[20] Crossbow, `www.xbow.com`.

[21] T. S. Datasheet, http://www.moteiv.com/products/docs/tmote-sky-datasheet.pdf.

[22] S. Croce, F. Marcelloni, and M. Vecchio, "Reducing power consumption in wireless sensor networks using a novel approach to data aggregation," *The Computer Journal*, vol. 51, no. 2, pp. 227–239, 2008.

[23] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proceedings of the IRE*, vol. 40, no. 9, pp. 1098 –1101, Sep. 1952.

[24] T. M. Cover and J. A. Thomas, *Elements of information theory.* New York, NY, USA: Wiley-Interscience, 1991.

[25] J. Kieffer, "A unified approach to weak universal source coding," *IEEE Transactions on Information Theory*, vol. 24, no. 6, pp. 674 – 682, Nov. 1978.

[26] A. Wyner, “On source coding with side information at the decoder,” *IEEE Transactions on Information Theory*, vol. 21, no. 3, pp. 294 – 300, 1975.

[27] D. Slepian and J. Wolf, “Noiseless coding of correlated information sources,” *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471 – 480, Jul. 1973.

[28] T. Cover, “A proof of the data compression theorem of slepian and wolf for ergodic sources (corresp.),” *IEEE Transactions on Information Theory*, vol. 21, no. 2, pp. 226 – 228, Mar. 1975.

[29] D. Marco and D. L. Neuhoff, “Reliability vs. efficiency in distributed source coding for field-gathering sensor networks,” in *IPSN*, 2004, pp. 161–168.

[30] C. E. Shannon, “Communication in the presence of noise,” *Proceedings of the IEEE*, vol. 72, no. 9, pp. 1192 – 1201, 1948.

[31] H. Nyquist, “Certain topics in telegraph transmission theory,” *Proceedings of the IEEE*, vol. 90, pp. 280 –305, 2002.

[32] A. V. Balakrishnan, “A note on the sampling principle for continuous signal,” *IRE Trans. Inform. Theory*, vol. 3, pp. 143–146, 1957.

CHAPTER II

Throughput Scaling in Random Wireless Networks: A Non-Hierarchical Multipath Routing Strategy

2.1 Introduction

The problem of asymptotic scalability of throughput in wireless networks has been investigated extensively under different assumptions on the network models. In their seminal work Gupta and Kumar [1] posed the problem of asymptotic scalability of random wireless networks and demonstrated that in dense networks a per-node throughput $\Omega(1/\sqrt{n \ln n})$ was achievable as the number of nodes in the network, n, goes to infinity. Equivalently, it can be said that *throughput scaling* of order $1/\sqrt{n \ln n}$ is attainable. It can be easily shown that this achievability result also holds for extended networks, that is, networks whose area is expanding such that the density of nodes remains constant. Further work in the area under different network models can be found in [2, 3, 4, 5, 6]. Recently Franceschetti et al[7] recently showed that this achievable per-node throughput may be increased. Specifically, they considered an extended network with approximately n randomly distributed nodes and multihop routing, and demonstrated that achievable per-node throughput can be increased to

$\Omega(\frac{1}{\sqrt{n}})$.

Compared to [1], the construction used in [7] introduced several new features. The first is a capacity-based link transmission rate formula as a function of the received signal-to-interference noise ratio (SINR), instead of the threshold-based binary rate model used in [1], where a positive bit-rate B is attainable when the SINR is above some threshold, and zero otherwise. (The former requires coding at each hop, while the latter does not.) The second is a routing hierarchy for data delivery in which data from a source is first delivered (via a single hop) onto a nearby highway – one of a system of communal highways, each with a horizontal and a vertical segment. The data is then multihopped along the highway (horizontally then vertically), and finally delivered from the highway to the destination in a single hop. By contrast, the method used in [1] is a simple shortest path type of routing, where a straight line is drawn connecting the source and the destination, and nodes along this line are selected to relay the data, forming an approximately straight line path. The third difference introduced in [7] is the use of percolation theory to construct the highways that serve as the main routing fabric in the network. Indeed, [7] is the first paper to use percolation theory to establish network throughput results.

The primary interest of the present chapter is to understand which of the above contribute to the increase in per-node throughput in a fundamental way, i.e., to understand the dependence of this new result on the above new features. The conclusion of this chapter is that the improved throughput scaling is principally due to the percolation-based routing, which enables shorter hops and, consequently, less interference. More precisely, the hops along the highways have bounded lengths that do not increase as the network expands. This would not have been possible if one were to use shortest path routing, the existence of which then invokes a connectivity requirement that would force the hop size to increase as the network expands. As another benefit this chapter shows an alternate construction to achieve the $\frac{1}{\sqrt{n}}$ throughput scaling.

This conclusion is established by showing that throughput $\Omega(\frac{1}{\sqrt{n}})$ can be attained by a system that does not employ highways, but rather uses percolation to establish, for each source-destination (s-d) pair, a set of $\Theta(\log n)$ disjoint routes within a narrow routing corridor running from source to destination. Thus with this multipath routing structure, highways and routing hierarchy are not essential. In addition, it is shown that throughput $\Omega(\frac{1}{\sqrt{n}})$ can be attained with the original threshold transmission bit-rate model, provided the transmission powers of the nodes are permitted to grow with n. Thus, the benefit of the capacity bit-rate model is simply to permit the power to remain bounded, even as the network expands. However, under the capacity bit-rate model the receiver's at the nodes have to be powerful in the sense that they need to be able to decode even when the SINRs are extremely low. Thus, using the the threshold based model asks for more powerful transmitters whereas the capacity bit-rate model asks for more powerful receivers. This emphasizes the importance of percolation based routing. Also, the multiplicity of routes, though each route can be dedicated to one s-d pair.

Better throughput may be attainable if one considers hierarchical cooperation [8], mobility [9, 10, 11], beamforming antennas [12] or hybrid networks [13, 14, 15] but it is nevertheless interesting to understand what throughput is attainable with simple systems that employing homogeneous node with omni-directional antennas that can only perform dumb relaying.

The remainder of the chapter is organized as follows, Section 2.2 introduces the system and the transmission rate models we use. Section 2.3 gives our main result and an overview of the proof. The formal proof follows in sections 2.4, 2.5, 2.6 and 2.7 which formalize the path construction, transfer rates, rates for delivery/draining and rates for intermediate hops, respectively. We finally finish the proof in 2.8 and provide concluding comments in 2.9.

2.2 Background and System Model

We consider a random extended network with nodes distributed over a disk $A_n \subset \mathcal{R}^2$, called the *network region*, whose radius $\sqrt{n}$ increases with n. We assume that nodes are placed randomly according to a Poisson point process of unit intensity over $\mathcal{R}^2$. Let $\mathcal{S}$ denote the set of node positions produced by the Poisson process, and let $\mathcal{S}_n$ denote the subset of nodes lying in A_n. Each node in $\mathcal{S}_n$ serves as a source of bits that it wishes to communicate to a destination, chosen randomly from the remaining nodes within the A_n. (A node may serve as a destination for more than one source.) Let $\mathcal{P}_n$ denote the set of s-d pairs, which we consider to constitute the random extended network. The number of nodes in $\mathcal{S}_n$, equivalently pairs in $\mathcal{P}_n$, is a random variable N_n with expected value πn. We are concerned with events that happen in the network region A_n with high probability, that is, probability tending to one as $n \to \infty$. We use the following version of the order notation, which is commonly used [7, 16] in analyzing asymptotic scalability. We will say that a non-negative sequence of random variables X_n is *probabilistically* $\Theta(y_n)$, or $\Theta_p(y_n)$ for short, when y_n is a non-negative sequence and there exist constants c_1 and c_2 such that $\Pr(c_1 y_n \leq X_n \leq c_2 y_n) \to 1$ as $n \to \infty$. *Probabilistically* $O(y_n)$, $O_p(y_n)$ and *probabilistically* $\Omega(y_n)$, $\Omega_p(y_n)$ are defined similarly. We also use the short hand *wpa one* if an event occurs with probability approaching one as n approaches infinity.

As shown in Lemma 12, N_n is probabilistically $\Theta(n)$. To simplify analysis, we assume that the nodes in $\mathcal{S}$ outside A_n are available for relaying data between a source and its destination, though we will only make use of nodes within $O(\ln n)$ of the disk. Note that since the radius of the disk is $\Theta(\sqrt{n})$, the disk outside the network region is extremely small.

Communication is done with multihop relaying in slots of some duration Δ seconds, which remains fixed throughout. There is a transmitter, receiver and omnidirectional antenna at each node. All transmitters use a common power P_n, which we get to choose and which may depend upon n. We assume that node j receives the transmission from node i with power $P_n \eta(d_{ij})$, where $\eta(\cdot)$ is a *propagation (or power attenuation) model* and d_{ij} is the Euclidean distance between nodes i and j. Gupta and Kumar [1] used the following propagation model,

$$\eta(d) = \frac{1}{d^\alpha}, \tag{2.1}$$

where $\alpha > 2$ is the propagation loss exponent. However, as $d \to 0$ the received power under this model grows to infinity. Hence, we refer to the above model as the *unbounded propagation model.* To avoid the unrealistic assumption of received power being higher than transmitted power, we use the *bounded propagation model*,

$$\eta(d) = \frac{e^{-\gamma d}}{(1+d)^\alpha} \tag{2.2}$$

where either $\gamma > 0, \alpha \geq 0$, or $\gamma = 0$ and $\alpha > 2$ are constants that depend on the channel conditions. For $\gamma = 0$ this model reduces to the one used by Dousse et al. [17] and Arpacioglu and Haas [18].

For any given system $\mathcal{P}_n$, the aim is design a communication protocol that allows each source to communicate bits to its destination at as high rate as possible. A simple protocol might be where each source communicates directly to its destination in a single hop. However, in this case simultaneous transmissions interfere at the receiver, resulting in a low rate. Thus, in order to improve the rate, we design a multi-hop system.

For a given network of s-d pairs $\mathcal{P}_n$, a *system* Σ_n is characterized by a set of *routes* and a *link schedule.* A route for a given *s-d* pair is a sequence of nodes from s to d. Each pair of successive nodes in a route is considered a *hop or link.* On each link, one node acts as a transmitter and the other as a receiver. A node may lie on more than one route and a pair of nodes may be a link on several different routes. The link schedule determines the links over which transmissions take place in a given time slot, the power at which the transmitters transmit and the number of bits transmitted over the links.

In designing a system, we first choose one or more routes for every $(s, d) \in \mathcal{P}_n$. For a given (s, d) pair, let $q_{i,s,d} = (l^1_{i,s,d}, l^2_{i,s,d}, ..., l^{m_i}_{i,s,d})$ denote its ith route, where $l^j_{i,s,d}$ is the jth link on the route. Let $Q(s, d) = \{q_{1,s,d}, q_{2,s,d}, ..., q_{m,s,d}\}$ denote the set of routes for a given s-d pair, and let $\mathcal{Q} = \{Q(s, d) : (s, d) \in \mathcal{P}_n\}$ denote the set of all routes in the system. Also, let $\mathcal{L}$ denote the set of all the links in the system. A link-schedule with period p_{ls}, denoted $\mathcal{LS}_{p_{ls}}$, is characterized by a sequence of sets $(LS_1, LS_2, ..., LS_{p_{ls}})$, where LS_k is a set of triplets, each indicating a link transmission in the kth slot. Specifically, (l, P, b) indicates that transmitter of link $l \in \mathcal{L}$ transmits b bits at power P with the goal of reaching the receiver for link l.

In the design approach used in this chapter and in [7], each route is composed of: a *draining* link from source to an intermediate node, a sequence of *intermediate or trunk* links between intermediate nodes and a *delivery* link from an intermediate node to the destination. Corresponding to the different types of links, we split the link schedule into three phases: draining, trunk and delivery. While designing the routes, we fix in advance an upper bound, d^{phase}, to the length of links in each phase. Fixing such an upper bound for each phase, allows us to fix a common power, P^{phase}, used for every transmission in that phase. As we will see later, this considerably simplifies obtaining an achievable bound on the number bits, b^{phase}, that can be successfully transmitted over links in a phase.

In order to design the link-schedule for one of these phases, we first design a cell-based periodic tdma schedule. We partition the network into square cells of fixed area and design the tdma schedule such that one node in each cell, transmitting at power P^{phase}, can successfully transmit b^{phase} bits to any receiver that is within distance d^{phase} of the transmitter in one time slot. The tdma schedule, denoted $\underline{T}_p^{phase}$, is characterized by power P^{phase}, number of bits b^{phase} and a sequence of sets $(T_1, T_2, ..., T_p)$, where T_k is the set of cells in which nodes can transmit in time slot k. Each set T_k is chosen such that the interference at each of the intended receivers is limited. This allows the transmission from each transmitting node to its receiver to be successful, i.e., the link transmission is successful. Since each cell may have more than one link originating from it, we construct the link-schedule by concatenating multiple periods of the tdma schedule. We refer to each period of the tdma schedule as an *epoch*. The number of epochs in the link-schedule is the maximum number of links that originate from a single cell. Then, every link in every cell is assigned precisely to one epoch of the tdma schedule. Indeed, if link l in cell ce is assigned to the kth epoch of the tdma schedule, then transmissions over l take place in the time slots assigned to ce in epoch k. Now, having designed the link-schedule for each of the three phases, the system link-schedule is a time-shared combination of these. That is, we concatenate a_{dr}, a_{tr} and a_{de} periods of draining, trunk and delivery trunk link-schedules, respectively, where the integers a_{dr}, a_{tr} and a_{de} are chosen so that the same number of bits are transmitted over each link within one period of the system link-schedule.

In order to evaluate the bits/slot that each source can transmit to its destinations, we will need a concept of link-bit potential and several notions of throughput. Let T be a set of simultaneously transmitting nodes (no schedule is presumed). The *link bit-potential* $BP_{i,j}$ is the maximum number of bits that can be successfully transmitted in one slot, when a node $i \in T$ transmits to a node j, not necessarily in T, in the presence of interfering transmissions from the other nodes in T.

We consider two models for link bit-potential based on received *signal to interference and noise ratio* (SINR). As in [1, 2, 5, 7] and many other papers, the SINR when node j receives data from node i in the presence of transmissions from the nodes in

T is

$$\mathrm{SINR}_{ij} = \frac{P_n\,\eta(d_{ij})}{N_0 + \sum_{\substack{k\in T\\ k\neq i}} P_n\,\eta(d_{kj})}\ ,$$

where N_0 represents the power in received background noise.

Capacity Link Bit-Potential Model: In this model, which was used in [7], the link bit-potential equals the capacity of an additive Gaussian channel with signal-to-noise ratio equal to $\mathrm{SINR}_{i,j}$, i.e.,

$$BP_{ij} = \frac{1}{2}\ln(1 + \mathrm{SINR}_{ij})\ , \tag{2.3}$$

Threshold Link Bit-Potential Model In this model, which has been more commonly used in throughput analysis of wireless networks [1, 19, 20, 21], the link-rate is

$$BP_{ij} = \begin{cases} B & \text{if } \mathrm{SINR}_{ij} \geq \tau \\ 0 & \text{else} \end{cases}\ , \tag{2.4}$$

where τ is some pre-determined threshold and $B > 0$ is a number less than the capacity given in (2.3).

We now give few different notions of throughput. Given a link-schedule $\mathcal{LS}$, a *throughput for a link*, is a rate in bits/slot, averaged over the link-schedule, at which bits get transmitted over the link, i.e., it is the link bit-potential multiplied by the number of times the link appears in the schedule divided by the period of the schedule. Note that the link-throughput for a given link l can either be evaluated for the phase in which l belongs or for the overall system schedule. Given a link-schedule $\mathcal{LS}$ for the system, the *throughput for a route* is the minimum of link-throughputs of all the links in the route. The *throughput of an s-d pair* is the sum of throughputs of all its routes. The *(overall, per-node) throughput* of a system, denoted $\mathcal{T}(\mathcal{P}_n, P_n, \mathcal{LS}_n)$, is the minimum throughput among all s-d pairs in $\mathcal{P}_n$.

In the seminal paper [1], it was shown that for a network where n nodes are distributed randomly in the network region A_n, a square of area 1, under the threshold link-rate and unbounded propagation models, for each n, there exists a way designing systems for $\mathcal{P}_n$ such that throughput is $\Omega_p(1/\sqrt{n \ln n})$. Since the network region was of area 1 the above was a dense network. However, it has been shown [21] that the same throughput is achievable in the case of expanding network where A_n is a circle of area n under the threshold link-rate model and the bounded or unbounded propagation model.

2.3 Main Result

The principal result of [7] is, essentially, that in extended networks, throughput $\Omega_p(1/\sqrt{n})$ is attainable when the capacity link bit-potential model is assumed. It was also shown that the power P_n can be set to a constant that does not depend on n or $\mathcal{P}_n$. The proof used a system of communal highways whose existence was demonstrated using percolation theory. The following theorem, which is the main result of this chapter, restates this result. However, we provide a proof that does not employ communal highways, though it does use a percolation approach similar to that in [7]. In addition, the theorem shows that the same throughput scaling is attainable with the threshold link bit-potential model, provided P_n is permitted to grow without bound.

Theorem 1 *Consider the bounded propagation model (2.2), either link bit-potential model, and for each n a random extended network $\mathcal{P}_n$ parametrized by n. There exists $b > 0$ and for each n there exists $P_n > 0$ and a collection Π_n of source-destination pair sets such that*

$$\Pr(\mathcal{P}_n \in \Pi_n) \to 1 \quad \textit{as } n \to \infty \ , \tag{2.5}$$

and for all sufficiently large n and each $\mathcal{P}_n \in \Pi_n$, there is a tdma schedule S_n that *when used with power P_n has throughput*

$$\mathcal{T}(\mathcal{P}_n, P_n, S_n) \geq \frac{b}{\sqrt{n}} \ . \tag{2.6}$$

Moreover, for the capacity link bit-potential model, P_n can be set to an arbitrary positive value P that does not vary with n, whereas for the threshold link bit-potential model, it is necessary and sufficient that P_n grows without bound as $n \to \infty$.

As mentioned in the introduction, one concludes from this theorem that it is percolation, rather than highways, that permit throughput scaling to increase from $\Omega_p(1/\sqrt{n \ln n})$ to $\Omega_p(1/\sqrt{n})$, and that the benefit of the capacity link bit-potential model is to permit the the power to remain constant, even as the network expands.

We now give an overview of the proof, details of which are in subsequent sections. We partition the design of the system into two parts. First we design the routes such that each route for each *s*-*d* pair has the proper hop lengths. We will then show that each link on each of these routes can sustain adequate link-throughput.

First, Section 2.4 shows, essentially, that for each source-destination pair $(s, d) \in \mathcal{P}_n$, one can find $\Theta_p(\ln n)$ nearly disjoint routes (the routes might have a few nodes in common), each composed of three segments and the following properties:[1]

1. The first segment is a single *draining hop* from the source to a node, called a *receptor*, on a *trunk* which is a sequence of short hops having nodes in the vicinity of both s and d. In its second segment, the route continues along the trunk in the direction of the destination until it reaches a node, called the *deliverer*, from which the destination can be reached in one *delivery hop*, which constitutes the final segment of the route.

[1] We said "essentially" because uniform bounds to the number of routes will actually be needed. Specifically, with M_n denoting the minimum number of routes found for an s-d pair in $\mathcal{P}_n$, then we actually need $M_n = \Theta_p(\ln n)$. The theorem and lemma statements to follow will make this clear. However, to streamline the discussion, we will not mention again the need for uniformity in paraphrased statements such as this.

2. The $\Theta_p(\ln n)$ trunks for (s, d) traverse a narrow, rectangular routing corridor of width $\Theta(\ln n)$ that contains both s and d. The trunk hop lengths are $O(1)$. Indeed, they are uniformly bounded by a constant independent of n, $\mathcal{S}_n$ and $\mathcal{P}_n$.

3. The draining and delivery hops have length $O(\ln n)$, due to the fact that the trunks, s and d lie within the routing corridor of width $\Theta(\ln n)$.

To demonstrate the attainability of throughput $\Omega_p(1/\sqrt{n})$, it will suffice to demonstrate that for any $\mathcal{P}_n$ there exists a link-schedule such that for each s-d pair in $\mathcal{P}_n$, each of its $\Theta_p(\ln n)$ routes sustain a route-throughput of $\Omega_p(1/(\sqrt{n}\ln n))$ for that pair. To do this we will show that each link in $\mathcal{Q}$ can sustain a link-throughput of $\Omega_p(1/(\sqrt{n}\ln n))$.

In the draining phase, each node sends its originating data a distance of $O(\ln n)$ to a receptor on each of the $\Theta_p(\ln n)$ trunks in its routing corridor. In order to calculate the link-throughput, we first find the rate, averaged over the tdma schedule at which a node can transmit to its receiver(s). We call this the *node-rate*. Theorem 5, which generalizes Theorem 3 of [7], will be used to show the existence of a tdma draining schedule that attains node-rate $\Omega(1/n^a)$[2], where $a < 1/2$, from each source to each of its receptors. In Section 2.6, using this tdma schedule, we construct the draining link-schedule under which each draining link can achieve a link-throughput of $\Omega(1/(n^a \ln n))$.

In the trunk phase, for each s-d pair, the trunk nodes relay data from the receptor to the deliverer with a sequence of hops of length $O(1)$. In this case, Theorem 5 can be used to show the existence of a tdma schedule such that each trunk node attains a node-rate of $\Omega(1)$. In Appendix 2.C that the maximum number of routes on which a given node can lie is $O_p(\sqrt{n}\ln n)$. Therefore, its $\Omega(1)$ node-rate needs to be shared

[2] No subscript $_p$ here.

among $O_p(\sqrt{n}\ln n)$ links. We can thus construct a trunk link-schedule under which each trunk link can sustain a link-throughput of $\Omega_p(1/\sqrt{n}\ln n)$.

Next, for the delivery phase, Lemma 9 and Corollary 10, are used to demonstrate the existence of a tdma delivery link-schedule such that each delivery-link can sustain a link-throughput $\Omega_p(1/(n^a(\ln n)^2)$.

Finally in Section 2.8 we combine the link-schedules for the three phases and show that every link can sustain a throughput of $\Omega_p(1/(\sqrt{n}\ln n)$. Thus, there are $\Omega_p(\ln n)$ routes for each source-destination pair, each sustaining a throughput of $\Omega_p(1/(\sqrt{n}\ln n))$. Therefore, each $(s,d) \in calP_n$ has a throughput $\Omega_p(\ln n) \times \Omega_p(1/(\sqrt{n}\ln n)) = \Omega_p(1/\sqrt{n})$, which is the desired result.

A summary of the remainder of this chapter follows. Section 2.4 uses percolation theory to construct suitable trunks; Section 2.5 presents the theorems for achievable node-rates; Section 2.6 establishes the phase link-throughputs attainable in the draining and delivery phases; Section 2.7 establishes the phase link-throughput attainable in the trunk phase; Section 2.8 completes the proof of the main result; and finally Section 2.9 makes concluding remarks.

2.4 Trunk Construction via Percolation

In this section, we show that as $n \to \infty$, for every s-d pair in $\mathcal{P}_n$ there exist $\Omega_p(\ln n)$ suitable trunks. By suitable we mean that the closest node on the path to the source on the trunk is $O(\ln n)$, likewise for the destination and each intermediate hop has length $O(1)$. Here the probability is taken over the random placement of the nodes using the Poisson point process and the random *s-d* assignments. We use a percolation approach similar to that used in [7] to establish the existence of highways. For more on percolation theory see [22] and [23].

Before going into details we give a brief overview of the trunk construction. First, we cover the the entire network region A_n with rectangular corridors whose dimensions are, roughly speaking, $\Theta(\sqrt{n}) \times \Theta(\ln n)$. The coverage of network region is such that for any two points in the network region there exists at least one corridor that contains both points. Note that as n increases these corridors become extremely narrow as compared to their length. Nevertheless, by individually tessellating each one of these corridors we show that there exist $\Omega_p(\ln n)$ disjoint path traversing the corridor lengthwise. Each path is a sequence of hops from one node to another, such that the length of each hop on each path is $O(1)$. By construction every s-d pair lies in at least one routing corridor, and the paths in one of these corridor will be used as trunks for the routes from s to d. Finally, we modify these paths slightly based on global tessellation, reasons for which will become clear in Section 2.5. While causing the paths to no longer be disjoint, this modification makes the calculation of attainable node-rate for each node on the trunk tractable.

We now describe the trunk construction in detail. Fix constants $c, \kappa > 0$ to be chosen later. Given n, consider a rectangle, illustrated in Figure 2.3(a), of length l_n and width w_n given by

$$\begin{aligned} l_n &= 2\sqrt{n} + \sqrt{2}c\epsilon_{1,n} \\ w_n &= \sqrt{2}c\left(\kappa \ln \frac{l_n}{\sqrt{2}c} + \epsilon_{2,n}\right), \end{aligned} \tag{2.7}$$

where $0 < \epsilon_{1,n}, \epsilon_{2,n} < 1$ are chosen such that $l_n/\sqrt{2}c$ and $w_n/\sqrt{2}c$ are integers. Note that, l_n is $\Theta(\sqrt{n})$ and w_n is $\Theta(\ln n)$. One of the narrow ends of the rectangle is labeled *left* and the other *right*. We wish to overlay the network region with a number of these rectangles such that for every pair of points in the network there exists a rectangle that contains both of them. We refer to such rectangles as *routing corridors*, as the trunks for any given s-d pair will mainly lie in one of these.

Divide the circumference of the network region A_n into $N_a = \left\lceil \frac{2\pi\sqrt{n}}{w_n} \right\rceil$ arc segments of length at most w_n as shown in Figure 2.1. For every pair of arcs, place a routing

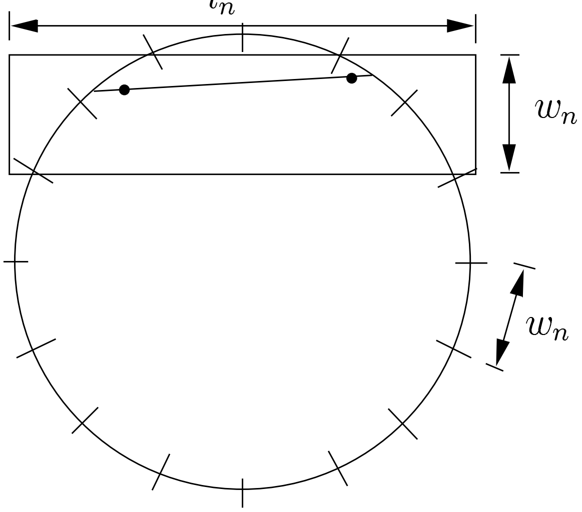


Figure 2.1: Routing corridors to cover the network region. The boundary of the network region is divided into arcs of size w_n. For each pair of arcs a corridor covering the two arcs is placed such that its center is as close to the center of network region as possible.

corridor such that the entire length of both arcs is contained in the corridor and that the center of the corridor is as close to the center of the network region as possible. Such a placement is always possible as the length of the arc segment at most equals the width of the corridor. Also, it can be easily seen that placing the center of the corridor closest to center of the network region uniquely specifies its location. And while this unique specification is not essential, it helps later in bounding the delivery and draining hop lengths.

Since there exists a routing corridor for every pair of arcs, the total number of routing corridors is

$$N_n = \frac{N_a(N_a-1)}{2} \leq \left(\frac{2\pi\sqrt{n}}{w_n}+1\right)^2 = O\left(\frac{n}{(\ln n)^2}\right) \ .$$

We claim that the above placement of routing corridors ensures that for any two points in the network region there exists at least one corridor that contains both of them. For any two points in the network region, extend the line joining them to

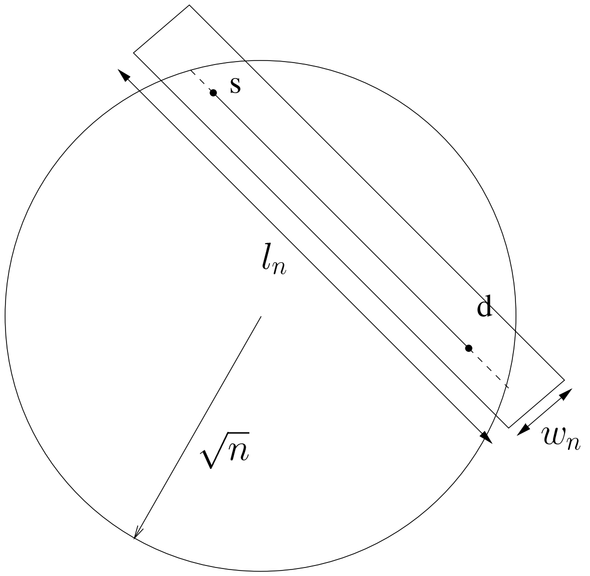


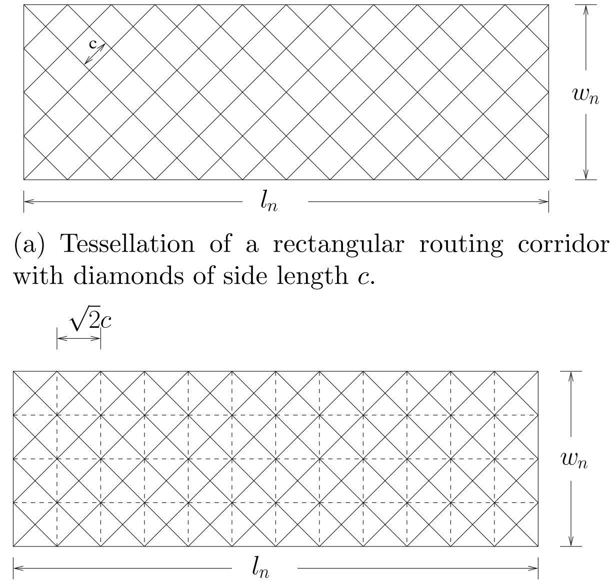
Figure 2.2: Routing corridor for a given s-d pair.

the intersect the boundary of A_n. This line will intersect two arc segments on the boundary. Since this pair of arc segments is contained in a routing corridor, so will the originally chosen points.

Closely following the percolation arguments in [7], we next show that each routing corridor contains $\Omega_p(\ln n)$ disjoint trunks. Tessellate each routing corridor with diamonds (i.e., rotated squares) with side length c, as is illustrated in Figure 2.3(a). Note that for any given diamond,

$$\Pr(\text{diamond contains at least one node}) = 1 - e^{-c^2} \triangleq p \ .$$

Draw horizontal *edges* across half the diamonds and vertical edges across the others in the manner shown with dotted lines in Figure 2.3(b). An edge is considered *open* if it lies in a diamond containing at least one node, and *closed* otherwise. A *path* is a sequence of connected edges, horizontal or vertical. A path is said to be *open* if it contains only open edges. Because each edge in an open path indicates the existence of a node within a diamond and because successive edges indicate nodes in adjacent



(a) Tessellation of a rectangular routing corridor with diamonds of side length c.

(b) Paths crossing the routing corridor from left to right are composed from horizontal and vertical edges, shown as dashed lines.

Figure 2.3: Routing corridor setup for s, d.

diamonds, an open path indicates the existence of a route with hops of length at most $\sqrt{5}c$, formed by taking one node from each diamond corresponding to each edge.

The theorem below shows that as $n \to \infty$ there are $\Omega_p(\ln n)$ disjoint open paths crossing any routing corridor lengthwise. With a small modification to be discussed later, these will form the basis of the trunks for the given (s, d) pair.

Assume that the corridors covering the network region are indexed 1 to N_n, and let $B_{n,i,m,c}$ denote the event that there exist at least m disjoint open c-paths that cross the ith routing corridor lengthwise. The following theorem, whose proof is contained in that of Theorem 5 of [7], is based on percolation theory.

Theorem 2 *Given $c > \sqrt{\ln 6}$ and $\kappa > 0$, then for any $\beta > 0$ and any integer i,*

$$\Pr(B_{n,i,m,c}) \geq 1 - \frac{4}{3}(6(1-p))^{\epsilon_{2,n}} \left(\frac{p}{1-p}\right)^{\beta\epsilon_{2,n}} \left(\frac{l_n}{\sqrt{2}c} + 1\right) \left(\frac{l_n}{\sqrt{2}c}\right)^{a} \tag{2.8}$$

where $m = \beta w_n$ and $a = \beta\kappa \ln\left(\frac{p}{1-p}\right) + \kappa \ln(6(1-p))$.

Note that part of the routing corridor will lie outside the network region and the paths whose existence the above theorem has demonstrated may make use of nodes that lie outside A_n. However, the final s-d routes do not make use of any nodes outside the network.

The following corollary uses the union bound to establish that as $n \to \infty$, there are $\Omega_p(\ln n)$ disjoint paths simultaneously in each one of the N_n corridors covering A_n.

Corollary 3 *Given $c > \sqrt{\ln 6 + 3/\kappa}$ and $\kappa > 0$, there exists a strictly positive constant $\beta(c, \kappa)$ such that*

$$\Pr(\cap_{i=1}^{N_n} B_{n,i,m,c}) \to 1 \quad \textit{as } n \to \infty \tag{2.9}$$

where $m = \beta(c, \kappa) w_n$.

Proof:

Since the number of routing corridors, N_n, is $O(n/(\ln n)^2)$, there exists b, n_0 such that, $N_n \leq bn/(\ln n)^2 \triangleq b_n$ for $n > n_0$. Then for $n > n_0$ and any $\beta > 0$,

$$\begin{aligned}
\Pr(\cap_{i=1}^{N_n} B_{n,i,m,c}) &\geq \Pr(\cap_{i=1}^{\lceil b_n \rceil} B_{n,i,m,c}) = 1 - \Pr(\cup_{i=1}^{\lceil b_n \rceil} B^c_{n,i,m,c}) \\
&\geq 1 - \sum_{i=1}^{\lceil b_n \rceil} \Pr(B^c_{n,i,m,c}) = 1 - \lceil b_n \rceil (1 - \Pr(B_{n,i,m,c})) \\
&\geq 1 - \lceil b_n \rceil \cdot \frac{4}{3}(6(1-p))^{\epsilon_{2,n}} \left(\frac{p}{1-p}\right)^{\beta\epsilon_{2,n}} \left(\frac{l_n}{\sqrt{2}c} + 1\right) \left(\frac{l_n}{\sqrt{2}c}\right)^a \\
&\geq 1 - \Bigg[\frac{4b}{3}(6(1-p))^{\epsilon_{2,n}} \left(\frac{p}{1-p}\right)^{\beta\epsilon_{2,n}} \\
&\qquad \left(\frac{l_n}{\sqrt{2}c} + 1\right) \left(\frac{l_n}{\sqrt{2}c}\right)^a \left(\frac{n}{(\ln n)^2} + 1\right)\Bigg],
\end{aligned}$$

where $m = \beta w_n$, the second inequality uses the union-bound, and the third uses Theorem 2 with a as defined in the theorem. Since l_n is $\Theta(\sqrt{n})$, the above expression goes to one as n tends to infinity if the exponent of n, $1/2+a/2+1$, is less than zero. This happens if and only if $\beta\kappa\ln(p/(1-p))+\kappa\ln(6(1-p))+3<0$. The facts that $\kappa>0$ and $c>\sqrt{\ln 6+3/\kappa}$ imply that $\kappa\ln(6(1-p))+3<0$. We can then choose $\beta=\beta(c,\kappa)$ small enough such that $\beta(c,\kappa)\kappa\ln(p/(1-p))+\kappa\ln(6(1-p))+3$ is less than 0. This results in $\Pr(\cap_{i=1}^{N_n}B_{n,i,m,c})\rightarrow 1$ as $n\rightarrow\infty$, completing the proof of the corollary. □

The next corollary uses the paths demonstrated in the previous corollary to construct routes for each s-d pair in $\mathcal{P}_n$.

Corollary 4 *Given $\kappa>0$ and $c>\sqrt{\ln 6+3/\kappa}$, there exists a strictly positive constant $\beta(c,\kappa)>0$ such that with probability approaching one for each s-d pair in $\mathcal{P}_n$ there exist $m=\beta(c,\kappa)w_n$ disjoint routes in some routing corridor such that the distances of the receptor from the source and deliverer from the destination are $O(\ln n)$ and every intermediate hop has length at most $\sqrt{5}c$.*

Proof: For any given s-d pair, consider the line joining the s,d positions and extend it to intersect the circumference of A_n. Consider the routing corridor that encompasses each of the arcs intersected by the line (see Figure 2.2). Give $\kappa>0, c>\sqrt{6+3/\kappa}$, according to Corollary 3, there exists $\beta(c,\kappa)$, such that there are $\beta(c,\kappa)w_n$ disjoint open paths that cross the routing corridor lengthwise. Now consider the part of the routing corridor that lies within the network region. Since there are $\beta(c,\kappa)w_n$ disjoint open paths that cross the routing corridor lengthwise, there will be $\beta(c,\kappa)w_n$ disjoint open paths in the truncated region as well. For each path in the routing corridor, designate the node closest to s on the truncated path as receptor and the node closest to d on the truncated path as the deliverer. Since the width of the routing corridor is w_n, it easily follows from the triangle inequality that the distance of the receptor from s is less than $w_n+\sqrt{2}c$, which is $O(\ln n)$. A similar argument holds for the distance of the deliverer from d. Also, the length of any intermediate hop on any s-d route is at most $\sqrt{5}c$. Thus, we show that for any s-d pair in $\mathcal{P}_n$ there exist $\beta(c,k)w_n$ routes, with distance of the receptor from s being $O(\ln n)$, likewise for d and the deliverer, and the length of each intermediate hop on each of these routes is at most $\sqrt{5}c$. $\square$

Before we end this section, we modify these routes slightly, so that relay nodes come from a single global tessellation of the entire network instead of a different tessellation for every routing corridor. As we will see in the next section, this makes the calculation of achievable node-rates tractable. Overlay the network region by a tessellation of squares of side c. If a square contains one or more nodes that lie in the network region, we designate one of them as the *relay node* for that square. For a square that contains no nodes that lie in the network region, no relay node is designated. Note that the distance of the designated relay node from any other node in the same square is at most $\sqrt{2}c$. Given any of the routes shown in Corollary 4, we modify it slightly by replacing any node i on the path that is not a designated relay node, by the designated relay node of the square that contains i. Using the triangle inequality we easily see that the length of the modified intermediate hop is at most $(\sqrt{5}+\sqrt{2})c$. Also, the draining and delivery hop lengths for the modified paths can be at most $(w_n+\sqrt{2}c)+\sqrt{2}c = O(\ln n)$. Note that while the original s-d routes were all disjoint, the modified routes may have some nodes in common. However, for any given s-d pair we can bound the number of routes on which any relay node lies. Any square in the global tessellation can at most intersect 9 squares of the tessellation of any given routing corridor. Thus a relay node may lie on at most 9 routes for that s-d pair. Finally, note that since w_n is $\Theta(\ln n)$, for large n there exists a constant ϕ such the the number of modified routes is at least $\phi \ln n$.

2.5 Attainable Node-Rates

As stated in the overview of the proof, we divide the time slots into three phases: draining, delivery and relay. We want to establish achievable link-throughputs for each one of these phases. In order to do this, we first establish *node-rates*, which are rates at which the nodes can successfully transmit to all receivers within a fixed distance via suitable tdma schedules. Then in the subsequent two sections we use these node-rates to find achievable link-throughputs. These will be established with the aid of several theorems presented here, which are variations and extensions of Theorem 3 and Corollary 1 of [7].

These theorems do not involve the sources, destinations or routes. Instead, here we try to establish the simultaneously achievable node rates when a set of transmitters satisfy a local density constraint and all receivers are at most distance d from their transmitters. The node-rate depends on the distance of the receiver from the transmitter and also the local density. A higher density implies either higher interference at the receiver if all the transmitters are transmitting or use of more time-slots if we want to time-share, both of which reduce achievable node-rate. We will subsequently use these theorems in order to find phase link-throughputs for the routes we designed in Section 2.4.

Tessellate $\Re^2$ with squares of side c. If there is one transmitting node in each cell, then this places a local density constraint as in any circle of area πc^2 the maximum number of transmitters is 9 and in a circle of area $2\pi c^2$ there is at least one transmitter. In order to limit interference at the receiver we time-share using a tdma schedule. Given a power $P > 0$ and link bit-potential r a *tdma schedule*, denoted $\mathcal{S}$, is a sequence of sets of cells $S_1, S_2,..., S_p$ such that during time slot i one node in each cell in S_i transmits at power P and link rate r.

Theorem 5 *Given $c > 0$, consider the tessellation of $\Re^2$ into $c \times c$ square cells with* *exactly one transmitting node in each square. Assume the bounded propagation model given by (2.2) with parameters α, γ and the capacity link bit-potential model (2.3) with noise power N_o. Then for any $P > 0$ there exists a constant $Z_C > 0$ depending on $c, \alpha, \gamma, P/N_o$ such that for any $d \geq 0$ there exists a link bit-potential r and a tdma schedule $\mathcal{S}$ under which each node achieves a node rate $Z_C \min\left\{1, e^{-\gamma d}/d^{2+\alpha}\right\}$ to all receivers that are within Euclidean distance d of the node.*

This theorem differs from Theorem 3 of [7] in that it is stated in terms of Euclidean distance between transmitters and receivers rather than counting the number of intermediate cells separating them. This leads to a different exponent. Moreover, it is stated in a way that makes it clear that there is a constant Z_C that applies for all $d \geq 0$. Note that the different transmitters may be transmitting to the same receivers. As long as the interference at the receiver is limited using the tdma schedule, the transmissions will be successful. The proof given below is based to a considerable degree on the proof in [7].

Proof: Given propagation model parameters α, γ, noise power N_o, a tessellation of $\Re^2$ into $c \times c$ square cells, and $P > 0$, let d_o denote the unique value such that $e^{-\gamma d_o}/d_o^{2+\alpha} = 1$, and let

$$\Psi \;=\; \sum_{i=1}^{\infty} \frac{i}{(i-1/2)^{\alpha}}\, e^{-\gamma c(i-1)} \tag{2.10}$$

Note that $0 < d_o \leq 1$ and that $\Psi < \infty$ when either $\gamma > 0$ or $\gamma = 0$ and $\alpha > 2$, which are the requirements for the propagation model. We will prove the theorem with

$$Z_C \;=\; \frac{1}{2}\frac{1}{\left(\frac{2}{c}+\frac{3}{d_o}\right)^2} \ln\left(1+\frac{1}{\frac{N_o}{P}+\frac{8\Psi}{2^{\alpha}}}\right) .$$

We first establish the attainability of node-rate $Z_C\, e^{-\gamma d}/d^{2+\alpha}$ for all $d \geq d_o$. Let us therefore fix $d \geq d_o$. As illustrated in Figure 2.4, partition $\Re^2$ into *super-squares*,

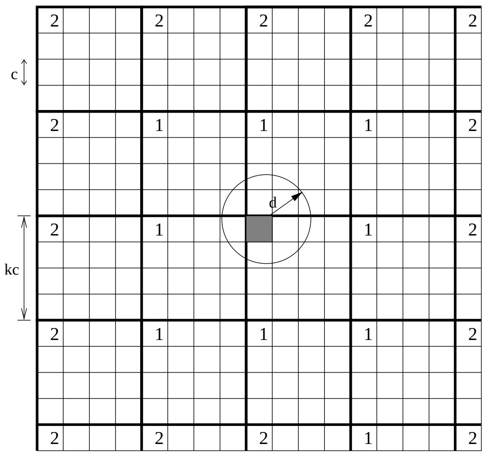


Figure 2.4: The partition into $kc \times kc$ super-squares, shown with $k = 4$. The shaded square in the center marks the cell where the transmitting node is located. The receiver is located within distance d of the transmitter. The cells marked "1" contain the eight nearest interfering transmitters. Those marked "2" contain the next 16 nearest interfering transmitters.

each composed of k^2 cells, where

$$k = \left\lceil \frac{2d}{c} + 2 \right\rceil \ .$$

Index the cells in each super-square in raster order. Consider the periodic tdma schedule with period k^2 such that in the ith slot of each period, from the cell in each super-square indexed by i, the transmitter in this cell transmits with power P at link bit-potential

$$\left(\frac{2}{c} + \frac{3}{d_o}\right)^2 Z_C \, \frac{e^{-\gamma d}}{d^\alpha} \ = \ \frac{e^{-\gamma d}}{d^\alpha} \frac{1}{2} \ln\left(1 + \frac{1}{\frac{N_o}{P} + \frac{8\Psi}{2^\alpha}}\right) \ . \tag{2.11}$$

We will show that at any receiver within distance d of one of the transmitters, the SINR is large enough to support the link bit-potential given in (2.11), i.e.,

$$\frac{1}{2} \ln(\mathrm{SINR} + 1) \ \geq \ \frac{e^{-\gamma d}}{d^\alpha} \frac{1}{2} \ln\left(1 + \frac{1}{\frac{N_o}{P} + \frac{8\Psi}{2^\alpha}}\right) \ . \tag{2.12}$$

Since the node-rate is the link bit-potential divided by k^2, it will directly follow that the attainable node-rate is at least

$$\begin{aligned}\left(\frac{2}{c}+\frac{3}{d_o}\right)^2 Z_C\,\frac{e^{-\gamma d}}{d^\alpha}\,\frac{1}{k^2} &\geq \left(\frac{2}{c}+\frac{3}{d_o}\right)^2 Z_C\,\frac{e^{-\gamma d}}{d^\alpha}\frac{1}{\left(\frac{2d}{c}+3\right)^2}\\ &\geq \left(\frac{2}{c}+\frac{3}{d_o}\right)^2 Z_C\,\frac{e^{-\gamma d}}{d^{2+\alpha}}\frac{1}{\left(\frac{2}{c}+\frac{3}{d_o}\right)^2}\\ &= Z_C\,\frac{e^{-\gamma d}}{d^{2+\alpha}} \qquad (2.13)\end{aligned}$$

which will complete the proof of the theorem.

Accordingly, as illustrated in Figure 2.4, consider a transmitter and receiver at most d apart in Euclidean distance. Note that there is one interfering transmitter in each cell of the other super-squares with the same index as the cell containing given transmitter. With P_S and P_I denoting, respectively, the power received from the intended and the interfering transmitters,

$$\text{SINR} = \frac{P_S}{N_0+P_I}\ . \qquad (2.14)$$

Since the Euclidean distance between the transmitter and receiver is at most d

$$P_S \geq P\eta(d) = P\min\left\{1, \frac{e^{-\gamma d}}{d^\alpha}\right\} = P\frac{e^{-\gamma d}}{d^\alpha}\ , \qquad (2.15)$$

where the last equality uses the facts that $d \geq d_o$ and $d_o \leq 1$.

Next, to bound the interfering power, note that, as illustrated in Figure 2.4, the closest 8 interferers to the receiver are at distance $kc-d-c$ or more, the next closest 16 interferers are at distance $2kc-d-c$ or more, and so on. The power from interfering nodes can thus be upper bounded as

$$\begin{aligned}
P_I &\leq \sum_{i=1}^{\infty} 8iP\eta(ikc-d-c) \;\leq\; \sum_{i=1}^{\infty} 8iP\frac{e^{-\gamma(ikc-d-c)}}{(ikc-d-c)^{\alpha}} \\
&= 8P\frac{e^{-\gamma d}}{c^{\alpha}k^{\alpha}}\sum_{i=1}^{\infty}\frac{i}{(i-\frac{d+c}{kc})^{\alpha}}e^{-\gamma kc(i-\frac{2d+c}{kc})} \\
&\leq 8P\frac{e^{-\gamma d}}{c^{\alpha}k^{\alpha}}\sum_{i=1}^{\infty}\frac{i}{(i-\frac{1}{2})^{\alpha}}e^{-\gamma c(i-1)} \;=\; 8P\Psi\frac{e^{-\gamma d}}{c^{\alpha}k^{\alpha}} && (2.16) \\
&\leq \frac{8P\Psi}{2^{\alpha}} && (2.17)
\end{aligned}$$

where the second inequality uses the formula for η, the third inequality uses our choice of k, including the fact that $k \geq 1$, and the fourth inequality again uses our choice of k, as well as the fact that $e^{-\gamma d}/d^{\alpha} \leq 1$ for $d \geq d_o$.

Substituting (2.15) and (2.17) into (2.14) gives

$$\text{SINR} \;\geq\; \frac{P\frac{e^{-\gamma d}}{d^{\alpha}}}{N_o + \frac{8P\Psi}{2^{\alpha}}} \;=\; \frac{e^{-\gamma d}}{d^{\alpha}}\frac{1}{\frac{N_o}{P}+\frac{8\Psi}{2^{\alpha}}}$$

It now follows that

$$\frac{1}{2}\ln(1+\text{SINR}) \;\geq\; \frac{e^{-\gamma d}}{d^{\alpha}}\frac{1}{2}\ln\left(1+\frac{1}{\frac{N_o}{P}+\frac{8\Psi}{2^{\alpha}}}\right)$$

This demonstrates (2.12) and completes the proof of the theorem for $d \geq d_o$.

Next, we show that for $d < d_o$ a rate of Z_C is attainable. As d decreases, the lower bound to P_S in (2.15) increases and the upper bound to P_I in (2.17) is the same as for $d = d_o$, thereby causing the lower bound to SINR to increase. Therefore for $d < d_o$, achievable node rate is at least as large as that for $d = d_o$ which is Z_C. This completes the proof of the theorem. □

The next theorem extends the previous theorem to the threshold link bit-potential model.

Theorem 6 *Given $c > 0$, consider the tessellation of $\Re^2$ into $c \times c$ square cells with exactly one transmitting node in each cell. Assume the bounded propagation model (2.2) with parameters α, γ, and the threshold link bit-potential model (2.4) with parameters by B, τ, N_o. Then there exists a constant $Z_T > 0$ depending on c, α, γ, τ, but not B, N_o, such that for any $d \geq 0$, there exists $P > 0$ and a tdma schedule $\mathcal{S}$ under which each node can attain a node rate $BZ_T \min\left\{1, \frac{1}{d^2}\right\}$ to all receivers within Euclidean distance d of the transmitter. The required power P increases to infinity as $d \to \infty$.*

Proof: Given propagation model parameters α, γ, given link bit-potential model parameters B, N_o, τ, and given a tessellation of $\Re^2$ into $c \times c$ square cells, let Ψ be defined as in (2.10). We will prove the theorem with

$$P \;=\; \frac{2\tau N_o}{\min\left\{1, \frac{e^{-\gamma d}}{d^\alpha}\right\}} \tag{2.18}$$

which grows to infinity as $d \to \infty$, and

$$Z_T \;=\; \frac{1}{\left(\max\left\{\frac{2}{c}+2, \frac{(16\Psi)^{1/\alpha}}{c}\right\}+1\right)^2}\;.$$

We first establish the attainability of node-rate BZ_T/d^2 for all $d \geq 1$. Let us therefore fix $d \geq 1$. As in the previous proof, consider a partition of $\Re^2$ into *super-squares*, each composed of k^2 cells, where

$$k \;=\; \left\lceil \max\left\{2\frac{d}{c}+2, \frac{(16\Psi)^{1/\alpha}}{c}, \frac{(16\Psi)^{1/\alpha}}{c}d\right\}\right\rceil\;. \tag{2.19}$$

As before, index the cells in each super-square in raster order, and consider the periodic tdma schedule with period k^2 such that in the ith slot of each period, from the cell in each super-square indexed by i, a transmitter transmits at link bit-potential

B with the power P specified in (2.18). The locations of the transmitters within their respective cells is arbitrary. To prove the theorem, we will show that at a receiver located within Euclidean distance d of any of the transmitters, the SINR is at least τ. It will then follow that with this schedule, the node-rate attained by each transmitter to such a receiver is B/k^2, which, we will show, is at least $BZ_T \min\left\{1, \frac{1}{d^2}\right\}$.

As in the previous proof, consider a transmitter and receiver at most distance d apart. Note that there is one interfering transmitter in each cells of the other super-squares with the same index as the transmitting cell. With P_S and P_I denoting, respectively, the power received from the intended and the interfering transmitters,

$$\text{SINR} \;=\; \frac{P_S}{N_0 + P_I} \;=\; \frac{1}{\frac{N_o}{P_S} + \frac{P_I}{P_S}} \;. \tag{2.20}$$

To show SINR $\geq \tau$, it suffices to show that both terms in the denominator of the right hand side are at most $1/(2\tau)$. Using (2.18), the first term in the denominator of (2.20) is

$$\frac{N_o}{P_S} \;\leq\; \frac{N_o}{P\eta(d)} \;=\; \frac{N_o}{P \min\left\{1, \frac{e^{-\gamma c d}}{d^\alpha}\right\}} \;=\; \frac{1}{2\tau} \;.$$

Next, using (2.19), (2.18) and (2.16), which uses $k \geq 2\frac{d}{c} + 2$, the second term in the denominator is

$$\frac{P_I}{P_S} \;\leq\; \frac{\frac{8P\Psi}{c^\alpha k^\alpha} e^{-\gamma d}}{P \min\left\{1, \frac{e^{-\gamma d}}{d^\alpha}\right\}} \;=\; \frac{\max\left\{\frac{16\Psi}{c^\alpha} e^{-\gamma d}, \frac{16\Psi}{c^\alpha} d^\alpha\right\}}{2k^\alpha} \;\leq\; \frac{1}{2\tau} \;.$$

Having shown that both terms in the denominator of (2.20) are at most $1/(2\tau)$, it follows that SINR $\geq \tau$, which implies that link bit-potential B is attainable. Therefore, the attainable node-rate is

$$\begin{aligned}
\frac{B}{k^2} &\geq \frac{B}{\left(\max\left\{2\frac{d}{c}+2, \frac{(16\Psi)^{1/\alpha}}{c}, \frac{(16\Psi)^{1/\alpha}}{c}d\right\}+1\right)^2} \\
&= \frac{B}{d^2}\frac{1}{\left(\max\left\{\frac{2}{c}+\frac{2}{d}, \frac{(16\Psi)^{1/\alpha}}{c}\frac{1}{d}, \frac{(16\Psi)^{1/\alpha}}{c}\right\}+1\right)^2} \\
&\geq \frac{B}{d^2}\frac{1}{\left(\max\left\{\frac{2}{c}+2, \frac{(16\Psi)^{1/\alpha}}{c}\right\}+1\right)^2} = \frac{BZ_T}{d^2}
\end{aligned}$$

where the second inequality uses $d \geq 1$. This completes the proof of the main part of this theorem.

Next we show that for $d < 1$ rate BZ_T is achievable. Note that as d decreases below 1, P_S increases and P_I can be lower bounded by the same quantity as for $d = 1$. Thus, the lower bound to SINR increases. This shows that rate BZ_T is achievable for $d < 1$.

It remains only to argue that P must increase to infinity as $d \to \infty$. If not, then the received power at distance d from a transmitter goes to zero as $d \to \infty$. Therefore, as can be seen from (2.20), SINR $\to 0$, and consequently, successful transmission is not possible for all sufficiently large d. □

The following corollaries to the previous two theorems, which corresponds to Corollary 1 to Theorem 3 in of [7], will be used to establish suitable node-rates for the delivery hop of routes. They follow simply by interchanging the roles of transmitter and receiver in the proofs of the two theorems.

Corollary 7 *Given $c > 0$, consider the tessellation of $\Re^2$ into $c \times c$ square cells with exactly one transmitting node in each cell. Assume the bounded propagation model given by (2.2) with parameters α, γ and the capacity link bit-potential model (2.3) with noise power N_o. Then for any $P > 0$ there exists a constant $Z_C > 0$ depending on $c, \alpha, \gamma, P/N_o$ such that for any $d \geq 0$ there exists a tdma schedule under which a* *receiver at an arbitrary point successfully receives data at rate* $Z_C \min\left\{1, e^{-\gamma d}/d^{2+\alpha}\right\}$ *from nodes transmitting, during the slots assigned to that cell, with power* P *at a suitable link bit-potential from any point within distance* d *of the receiver.*

Corollary 8 *Given* $c > 0$*, consider the tessellation of* $\Re^2$ *into* $c \times c$ *square cells with exactly one transmitting node in each cell. Assuming the bounded propagation model (2.2) with parameters* α, γ*, and the threshold link bit-potential model (2.4) with parameters by* B, τ, N_o*. Then there exists a constant* $Z_T > 0$ *depending on* c, α, γ, τ*, but not* B, N_o*, such that for any* $d \geq 0$*, there exists* $P > 0$ *and a tdma schedule under which a receiver at an arbitrary point successfully receives data at rate* $BZ_T \min\left\{1, \frac{1}{d^2}\right\}$ *from nodes transmitting, during the slots assigned to that cell, with power* P *at a link bit-potential* B *from any point within distance* d *of the receiver. The required power* P *increases to infinity as* $d \to \infty$*.*

Before we end the section, we make note that in all the above theorems and corollaries we assume that each cell contains exactly one node. However, it should be easy to see in the scenario where some nodes are empty, the rates in the theorems, and hence the corollaries, would still be achievable as the signal and interfering power can still be bounded by the same quantities as in the proof of the theorems.

2.6 Link-Throughputs: Draining and Delivery

In this section, we establish the link-throughputs for the draining and delivery phases. This is done by establishing link-schedules for the two phases by using the tdma schedules established in the theorems and corollaries in the previous section.

We first consider the draining phase. Since every node in the network acts as a source and no relaying takes place during this phase, the entire node-rate at the source during the delivery phase is devoted to transferring its own data to all its receptors.

Lemma 9 *For each n, let $\mathcal{P}_n$ be a random extended network on A_n, and for each $(s,d) \in \mathcal{P}_n$ let there be m routes from s to d, each having its receptor within distance a_n of the source. Then under the bounded propagation model (2.2) there exist constants Z_C, Z_T and with probability approaching one as $n \to \infty$ there exists a link-schedule depending on n such that the link-throughput for every draining links is*

$$T_{dr}(a_n) = \begin{cases} Z_C \min\left\{1, \dfrac{e^{-\gamma a_n}}{a_n^{\alpha+2}}\right\} \dfrac{1}{m \ln n} & \textit{under the capacity link bit-potential model} \\ B Z_T \min\left\{1, \dfrac{1}{a_n^2}\right\} \dfrac{1}{m \ln n} & \textit{under the threshold link bit-potential model} \end{cases}$$

Proof: Let $c > 0$. First, for the capacity link bit-potential model, consider an overlay of A_n with a tessellation of squares of side length c. Let $\mathcal{S}_n = (S_1,..., S_p)$ be the tdma schedule allotting time slots to the cells in the tessellation established using Theorem 5 for nodes transmitting over distance a_n. In order to construct the draining link schedule we need to know the maximum number of links originating in a cell.

Let N_i be the number of nodes in a cell and let $N_{max} = \max_i N_i$. Since each node has m receptors, there are mN_i links originating out of the ith cell. Label them $1,..., mN_i$ and let L_i be the set of links labeled i in all the cells. In order to assign each link at least one transmitting slot in the link-schedule we need mN_{max} periods of the tdma schedule. Consider the draining link schedule

$$LS = \left((\mathcal{S}_n^1, L_1), (\mathcal{S}_n^2, L_2),..., (\mathcal{S}_n^{mN_{max}}, L_{mN_{max}})\right),$$

where $\mathcal{S}_n^i = \mathcal{S}_n$ is called the i epoch of the tdma schedule. Transmissions over the links labeled i take place in the ith epoch of the tdma schedule.

The node-rate at which one node in each cell can transmit to all receivers located within distance a_n given by Theorem 5 is $R(a_n) = Z_T \min\{1, e^{-\gamma a_n} a_n^{-\alpha-2}\}$. Since transmissions over a link take place in precisely in one epoch of the tdma Schuyler and since there are mN_{max} total epochs, the link-throughput is

$$T_{dr}(a_n) = Z_C \min\left\{1, \frac{e^{-\gamma a_n}}{a_n^{\alpha+2}}\right\} \frac{1}{mN_{max}}$$

From Lemma 13 in the appendix, we know that with probability approaching one $N_{max} \leq \ln n$, where the maximum is over all cells in the network region. Thus, with probability approaching one as n approaches infinity, the link-throughput during draining phase is,

$$T_{dr}(a_n) = Z_C \min\left\{1, \frac{e^{-\gamma a_n}}{a_n^{\alpha+2}}\right\} \frac{1}{m \ln n}$$

The proof for the threshold based model follows a similar argument. □

The following corollary, which is a dual of the previous lemma, gives the link-throughput at which a destination can receive data from its deliverers during the delivery phase.

Corollary 10 *For each n, let $\mathcal{P}_n$ be a random extended network on A_n, and for each $(s,d) \in \mathcal{P}_n$ let there be m routes such that the deliverer is within distance a_n of the destination. Then under the bounded propagation model (2.2) there exist constants Z'_C, Z'_T and with probability approaching one as $n \to \infty$ there exists a link schedule depending on n such that each destination can receive from each one of its deliverers at link-throughput*

$$T_{de}(a_n) = \begin{cases} Z'_C \min\left\{1, \dfrac{e^{-\gamma d_n}}{a_n^{\alpha+2}}\right\} \dfrac{1}{a_n^2 \ln n} & \textit{under the capacity link bit-potential model} \\ BZ'_T \min\left\{1, \dfrac{1}{a_n^2}\right\} \dfrac{1}{a_n^2 \ln n} & \textit{under the threshold link bit-potential model} \end{cases}$$

Proof: Let $c > 0$. First, for the capacity link bit-potential model, consider an overlay of A_n with a tessellation of squares of side length c. Let $L_{de,i}$ denote the number of delivery links the originate in cell i. Label these links $1, 2, ..., L_{dr,i}$. Let $L_{de} = \max_i L_{de,i}$, where the maximum is taken over all cells. Let $\mathcal{S}_n = (S_1, ..., S_p)$ be the tdma schedule allotting time slots to the cells in the tessellation established using Theorem 5 for nodes transmitting over distance a_n. To construct the link-schedule for delivery, consider L_{de} periods of this tdma schedule. In the ith period of the tdma schedule the delivery node in each cell transmits over the link labeled i in all time slots that are assigned to it. The node-rate at which one node in each cell can transmit to all receivers located within distance a_n given by Theorem 5 is $R(a_n) = Z_T \min\{1, e^{-\gamma a_n} a_n^{-\alpha-2}\}$. Since transmissions over a link take place in precisely in one period of the tdma schedule and since there are L_{de} total periods, the link-throughput is

$$T_{de}(a_n) = Z_C \min\left\{1, \frac{e^{-\gamma a_n}}{a_n^{\alpha+2}}\right\} \frac{1}{L_{de}}$$

Next we upper bound L_{de}, by the product of maximum number of nodes within distance a_n of each cell and the maximum number of nodes that can choose a given node as destinations. Since the distance between a destination and its deliverer is at most a_n, using the Chernoff bound it can be shown that there exists δ such that with probability approaching one as n approaches infinity there are at most δa_n^2 nodes within distance a_n of the deliverer. Using the Chernoff bound it can also be shown that the there exists δ' such that with probability approaching one as n approaches infinity the number of sources that choose a give node as destination is at most $\delta' \ln n$. Thus, with probability approaching one as n approaches infinity, L_{de} is less than $\delta\delta' a_n^2 \ln n$.

Thus, with probability approaching one as n approaches infinity the link-throughput for the delivery phase is

$$T_{de}(a_n) = Z'_C \min\left\{1, \frac{e^{-\gamma a_n}}{a_n^{\alpha+2}}\right\} \frac{1}{a_n^2 \ln n},$$

where $Z'_C = Z_C/(\delta\delta')$.

The proof for threshold link bit-potential model follows a similar argument. □

We now establish achievable link-throughputs for the draining and delivery phases for the routes created in Section 2.4. We do this for the capacity link bit-potential model. The arguments for threshold rate-model are similar. Recall that with the global tessellation of A_n into squares of side c, all receptors are within distance $w_n + 2\sqrt{2}c$ of the source where w_n is as given in (2.7). From Lemma 9, the link-throughput for the draining phase is

$$Z_C e^{-\gamma(w_n+2\sqrt{2}c)} \frac{1}{(w_n + 2\sqrt{2}c)^{\alpha+2}} \frac{1}{m \ln n} \geq Z'_C \frac{n^{-\gamma c\kappa/\sqrt{2}}}{m \ln n^{\alpha+3}} \ ,$$

where $Z'_C = Z_C e^{-2\sqrt{2}\gamma c}(\sqrt{2}c/3)^{2\sqrt{2}\gamma c\kappa}(2\sqrt{2}c(\kappa+1))^{-\alpha-2}$. As shown in Corollary 3, with probability approaching one as n approaches infinity there are $\phi \ln n$ routes for every s-d pair. Thus, with probability approaching one, during the draining phase the link-throughput is $Z''_C n^{-\gamma c\kappa/\sqrt{2}}(\ln n)^{-\alpha-4}$ where $Z''_C = Z'_C/(\phi)$. It can be similarly shown that there exists Z'''_C such that link-throughput $Z'''_C n^{-\gamma c\kappa/\sqrt{2}}(\ln n)^{-\alpha-5}$ is achievable for the delivery phase.

Finally, if $\gamma c\kappa/\sqrt{2} < 1/2$, then for large n there exists $Z_{C,1}, a < 1/2$ such that link-throughput $Z_{C,1}n^{-a}$ is achievable for both draining and delivery phases. Also, note that the length of link-schedule for draining and delivery phases are $O((\ln n)^2)$ and $O((\ln n)^3)$ respectively.

2.7 Link-Throughput: Trunk Phase

In this section, we want to establish achievable link-throughput for the trunk phase. We proceed in the same way as for draining and delivery phases. We first establish the tdma schedule such that one relay node in each cell can transmit to each one of its neighboring relay nodes. We calculate the node rate for this tdma schedule. Using this tdma schedule and the maximum number of trunk-links that originate in a cell construct the link-schedule for trunk phase.

Recall from Section 2.4 that with a global tessellation if A_n into squares of side c, two neighboring relay nodes are at most distance $(\sqrt{2}+\sqrt{5})c$ apart. Now from Theorems 5 and 6 it can be easily seen that each relay node can transmit to every one of its neighboring relay nodes at node-rates

$$\begin{aligned} Z_{C,tr} &\triangleq \min\left\{1, \frac{e^{-\gamma(\sqrt{2}+\sqrt{5})c}}{((\sqrt{5}+\sqrt{2})c)^{\alpha+2}}\right\} && \text{under capacity-link rate model} \\ Z_{T,tr} &\triangleq \min\left\{1, \frac{1}{((\sqrt{5}+\sqrt{2})c)^{2}}\right\} && \text{under threshold link bit-potential model .} \end{aligned}$$

Note that $Z_{C,tr}$ and $Z_{T,tr}$ do not depend on n. Also, the length of tdma schedule is a constant, that is, it does not depend on n.

We next upper-bound the number of trunk-links that originate in a cell. Note that number of trunk-links originating in a cell is same as the number of routes on which the designated relay node for that cell lies. We refer to this number as the *loading factor* of the relay node or alternatively the loading factor of the square containing the relay node. Let M_n denote the number of squares in the global tessellation of A_n. It is easy to see that for large n, $M_n \le \left\lceil \frac{2\sqrt{n}}{c} \right\rceil^2 \le \frac{9n}{c^2}$. Number the squares 1,..., M_n, and let $L_i(n)$ denote the loading factor of the ith square. Also, let $L(n) = \max_i L_i(n)$. We observe that if an s-d pair contributes a path or paths to the $L_i(n)$, then it must be that the routing corridor for s-d intersects the ith square.

Lemma 11 *For a tessellation of the network region into squares of side c, there exists a constant δ such that*

$$\Pr(L(n) \leq \delta\sqrt{n}\ln n) \to 1 \text{ as } n \to \infty \ .$$

The proof for the lemma is given in Appendix 2.C.

To get an achievable link throughput for the trunk phase consider the link-schedule that is concatenation of $L(n)$ periods of the tdma schedule. During the ith period of the link schedule the designated relay node transmits over the ith link originating in the node, in the times slots assigned to its cell. Since the node-rate under the tdma schedule is $Z_{C,1}$ and since each link gets transmitted over in precisely one of the tdma schedule periods, an achievable link-throughput is $Z_{C,1}/L(n)$. Using Lemma 11, with probability approaching one as n approaches infinity link-throughput $Z_{C,tr}/(\delta\sqrt{n}\ln n) = Z'_{C,tr}/(\sqrt{n}\ln n)$ is achievable for the trunk phase.

Following similar arguments, an achievable link-throughput under the threshold rate model is $Z'_{T,1}/(\sqrt{n}\ln n)$. Finally, note that the length of the link schedule is $O(\sqrt{n}\ln n)$.

2.8 Completion of Proof

In this section, we establish a link-schedule for the system and show that this link-schedule achieves a throughput of $\Omega_p(1/\sqrt{n})$. We construct this link-schedule for the routes chosen in Section 2.4 and make use of draining, delivery and trunk link-schedules that were constructed in Sections 2.6 and 2.7.

If $\gamma > 0$, choose $c > 3\sqrt{2}\gamma + \sqrt{18\gamma^2 + \ln 6}$ and $\kappa = \frac{1}{2\sqrt{2}c\gamma}$. Else if $\gamma = 0$, choose $\kappa = 1$ and $c > \sqrt{\ln 6 + 3}$. Note that in both cases $\gamma c\kappa/\sqrt{2} < 1/2$.

For each n, consider an overlay of A_n with a tessellation of squares of side c. Since c and κ satisfy the conditions in Corollary 3, from Section 2.4, w. p. a. one there exist $\phi \ln n$ routes for every s-d pair in $\mathcal{P}_N$ such that the receptor and deliverer on each route is within distance $w_n + 2\sqrt{2}c$ of the source and destination respectively, and each intermediate hop is of length at most $(\sqrt{2}+\sqrt{5})c$.

In Sections 2.6 and 2.7 we constructed draining, delivery and trunk link-schedules of length $O_p((\ln n)^2)$, $O_p((\ln n)^3)$ and $O_p(\sqrt{n}\ln n)$ respectively. Thus, there exists a δ_1 such that w. p. a. one the lengths of these schedules are less than $\delta_1(\ln n)^2$, $\delta_1(\ln n)^3$ and $\delta_1\sqrt{n}\ln n$, respectively. Also w. p. .a. one, the number of bits that can be transmitted over each link in one period is $Z_{C,1}n^{-a}$ over draining and delivery schedule and $Z_{C,tr}$ over the trunk schedule. Now in each period of the overall link-schedule, we want the same number of bits to be transmitted over each link. Thus, consider the overall link-schedule that is a concatenation of $\lceil n^a \rceil, 1$ and $\lceil n^a \rceil$ periods of the draining, trunk and delivery link-schedules, respectively. Then the length of the overall link schedule is $L \leq \delta_1(n^a(\ln n)^2 + n^a(\ln n)^3 + \sqrt{n}\ln n)$. Since $a < 1/2$, for large n, $L < \delta_2\sqrt{n}\ln n$, where $\delta_2 = \delta_1 + 1$. It should be easy to see that w. p. a. one in this schedule $Z = \min\{Z_{C,1}, Z_{C,tr}\}$ bits can be transmitted over one period. Thus, with probability approaching one as n approached infinity, throughput $\frac{Z}{\delta_2\sqrt{n}\ln n}$ is achievable for every link and consequently for every route. Since, with probability approaching one as n approaches infinity, there are $\phi \ln n$ routes for every $(s, d,) \in \mathcal{P}_n$, the throughput of the system is $\frac{Z}{\delta_2\sqrt{n}\ln n} \times \phi \ln n = \Omega_p\left(\frac{1}{\sqrt{n}}\right)$. This completes the proof of the main result.

2.9 Concluding remarks

In conclusion, the gain in throughput in Franceschetti et al. [7] made over Gupta and Kumar [1] come from using percolation to reduce the length of intermediate relay hops to a constant distance as compared to the Gupta and Kumar version where all hop lengths were $O(\ln n)$. This allows for more simultaneous transmissions to take place throughout the network during the relaying phase and improves the relaying rate.

The gain from using the capacity link-rate model is that power at the transmitters can now doesn't have to increase with n as it has to with the threshold link-rate model. However, using the capacity link-rate model means that the receivers, especially during draining and delivery slots, have to be able to receive at SINRs that approach zero. Thus, under the threshold link-rate model we need a powerful transmitter whereas under the capacity link-rate model we need a powerful receiver.

2.A Number of nodes in the network region

The follow lemma shows that the number of nodes, denoted N_n, in the network region A_n, is $\Theta_p(n)$. That that $N_n = \Omega_p(n)$ is used later when applying the union bound.

Lemma 12

$$\Pr(\frac{\pi n}{2} < N_n < 2\pi n) \to 1 \quad \textit{as } n \to \infty \ . \tag{2.21}$$

The probability that the number of nodes, N_n, in the network region A_n is between $\pi n/2$ and $2\pi n$ and goes to 1 as $n \to \infty$.

Proof: The number of nodes in the network region, N_n, is a Poisson random variable with mean πn. Applying the Chernoff bound gives

$$\Pr(N_n > 2\pi n) \leq e^{-2s\pi n}\,\mathsf{E}[e^{sN_n}] = e^{-2s\pi n}e^{\pi n(e^s-1)}$$

for all $s > 0$. Choosing $s = 1$ gives

$$\Pr(N_n \leq 2\pi n) \geq 1 - e^{-2\pi n}e^{\pi n(e-1)}$$
$$1 - e^{\pi n(3-e)} \to 1 \ \text{ as } n \to \infty \ .$$

Applying the Chernoff bound again gives $\Pr(N_n > \pi n/2) \to 1$.

$\square$

2.B Number of nodes in a each square of tessellation of A_n

In this appendix we find an upper bound to the number of nodes in the global tessellation of A_n into square cells of side c.

Lemma 13 *Given a tessellation of network region A_n into squares of side c, let N_i denote the number of nodes in cell i, and let $N = \max_i N$. If the nodes in the network region are placed according to a Poisson process of unit intensity, then*

$$\Pr(N < \ln n) \to 1 \qquad n \to \infty$$

Proof:

Let M_n be the number of squares in the tessellation of A_n. Since A_n is contained in a square of side $2\sqrt{n}$, it is easy to see that for all sufficiently large n

$$M_n \leq \left\lceil \frac{2\sqrt{n}}{c} \right\rceil^2 \leq \left(\frac{2\sqrt{n}}{c} + 1 \right)^2 \leq \frac{9n}{c^2}\ . \tag{2.22}$$

For any square i the number of nodes, N_i, is a Poisson distributed random variable with mean c^2. Applying the Chernoff bound gives,

$$\Pr(N_i \geq \ln n) \leq e^{-2s \ln n}\, \mathsf{E}[e^{sN_i}] = e^{-2s \ln n} e^{c^2(e^s - 1)} \tag{2.23}$$

for all $s > 0$. Thus,

$$\begin{aligned}\Pr(N \geq \ln n) = \Pr(\max_i N_i \geq \ln n) &\leq M_n \Pr(N_1 \geq \ln n) \\ &\leq \frac{9n}{c^2} \frac{e^{c^2(e-1)}}{n^2} \to 0 \qquad n \to \infty,\end{aligned}$$

where the first inequality follows from union bound and the second inequality follows from (2.22) and (2.23) with $s = 1$.

□

2.C Loading Factor

In this section, we provide a proof of Lemma 11. That is, we provide an upper bound to the loading factor of every square cell in the global tessellation of A_n.

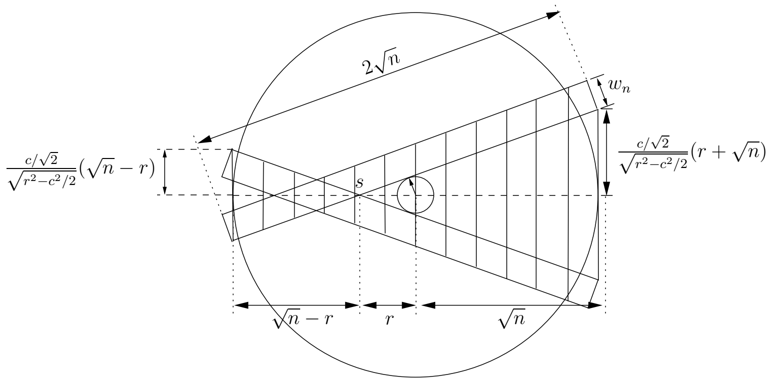
Proof of Lemma 11:

Recall that L_i is the loading factor of the ith cell and $L(n) = \max_i L_i$. For a fixed δ to be chosen later and any n,

$$\begin{aligned} \Pr(L(n) \leq \delta\sqrt{n}\ln n) &= \Pr\left(\max_i L_i \leq \delta\sqrt{n}\ln n\right) \\ &\geq 1 - \sum_{i=1}^{M_n} \Pr(L_i > \delta\sqrt{n}\ln n) \end{aligned} \tag{2.24}$$

where $M_n \leq \frac{9n}{c^2}$ is the number of cells in the network region. Let $A_{ij} = 1$ if the corridor for the jth (s,d) pair intersects the ith cell and zero otherwise. Recall that $L_i(n)$ is the number of routes that contain the ith cell and also, as shown in Section 2.4, that if the corridor for an (s,d) pair intersects cell i, then cell i may be part of at most nine routes for that (s,d) pair. Thus, $L_i \leq \sum_{j=1}^{N_n} 9A_{ij}$. Note that for a given i, A_{i1}, A_{i2}... are independent and identically distributed. However the L_i's are not identically distributed. Instead L_i will generally have a higher value for squares near the center of A_n than its boundary. Let $p_{i,j}$ be the probability that $A_{i,j} = 1$. Then let $a \triangleq \delta\sqrt{n}\ln n$, applying the Chernoff bound,

$$\begin{aligned} \Pr(L_i(n) > a) &\leq \min_{s>0} e^{-sa}\,\mathsf{E}[e^{sL_i(n)}] \leq \min_{s>0} e^{-sa}\,\mathsf{E}\left[e^{9s\sum_{j=1}^{N_n} A_{ij}}\right] \\ &= \min_{s>0} e^{-sa} \sum_{k=0}^{\infty} \Pr(N_n = k)\,\mathsf{E}\left[e^{9s\sum_{j=1}^{k} A_{ij}}\right] \end{aligned}$$



$2\sqrt{n}$
w_n
$\frac{c/\sqrt{2}}{\sqrt{r^2-c^2/2}}(r+\sqrt{n})$
$\frac{c/\sqrt{2}}{\sqrt{r^2-c^2/2}}(\sqrt{n}-r)$
s
$\sqrt{n}-r$
r
$\sqrt{n}$

Figure 2.5: $\Delta(s,d) \le c/\sqrt{2} + w_n$ if and only if the destination lies in the striped region. The radius of the circle at origin is $c/\sqrt{2}$.

$$= \min_{s>0} e^{-sa} \sum_{k=0}^{\infty} \Pr(N_n = k) \prod_{j=1}^{k} [p_{i,j} e^{9s} + (1 - p_{i,j})] \tag{2.25}$$

In order to make the evaluation of the above bound tractable, we want a uniform upper bound on $p_{i,j}$.

Lemma 14 *For sufficiently large n, there exists μ such that*

$$p_{i,j} \le p_n \triangleq \mu \frac{\ln n}{\sqrt{n}} \text{ , } \textit{for all } i, j \text{ .} \tag{2.26}$$

Proof: We setup a polar coordinate system such that the origin lies at the center of the network region. As the probability of intersection of a square by a random s-d pair routing corridor is highest at the center, we consider the ith square to lie at the center of the network region, i.e., to contain the origin. Since such a square of side c is completely contained in a circle of radius $c/\sqrt{2}$, we upper bound $p_{i,j}$ by the probability of the routing corridor for (s,d) intersecting a circle of radius $c/\sqrt{2}$ centered at the origin.

We place the source, s, at a point (r,θ) and find an upper bound to the probability that the routing corridor for this s intersects the circle at the origin and then we average over r and θ. Let $\Delta(s,d)$ denote the distance from the origin to the line joining s and d. Since the routing corridors have width w_n, a routing corridor for an (s,d) pair can intersect the circle at the origin only if $\Delta(s,d) \leq c/\sqrt{2} + w_n$. Thus,

$$\begin{aligned}&\Pr(s\text{-}d \text{ routing corridor intersects cell } \mid s = (r,\theta))\\ &\leq \Pr(\Delta(s,d) \leq c/\sqrt{2} + w_n \mid s = (r,\theta))\end{aligned}$$

For $r > c/\sqrt{2} + w_n$, $\Delta(s,d) \leq c/\sqrt{2} + w_n$ if and only if the destination lies in the vertically striped region in Figures 2.5.

$$\begin{aligned}&\Pr\left(\text{s-d routing corridor intersects square i} | s = (r,\theta)\right)\\ &\leq \begin{cases} \frac{4\delta_1 \ln n}{\pi\sqrt{n}} + \frac{(c/\sqrt{2})}{\sqrt{r^2-(c/\sqrt{2})^2}}\frac{(r+\sqrt{n})^2}{\pi n} + \frac{(c/\sqrt{2})}{\sqrt{r^2-(c/\sqrt{2})^2}}\frac{(\sqrt{n}-r)^2}{\pi n} & \text{if } r > c/\sqrt{2} + w_n \\ 1 & \text{if } r \leq \frac{c}{\sqrt{2}} + w_n \end{cases},\end{aligned}$$

where the upper bound is easily obtained by calculating the areas of the striped region in Figure 2.5. Since the joint probability density of the polar coordinate locations is $p(r,\theta) = \frac{2r}{n}\frac{1}{2\pi}$, we have

$$\begin{aligned}p_{n,i} &= \int_0^{2\pi}\int_0^{\sqrt{n}} \Pr\Big(\begin{matrix}\text{s-d routing corridor}\\ \text{intersects square } i\end{matrix}\Big| s = (r,\theta)\Big) p(r,\theta)\, dr d\theta \\ &\leq \int_0^{\frac{c}{\sqrt{2}}+w_n} \frac{2r}{n} dr + \int_{\frac{c}{\sqrt{2}}+w_n}^{\sqrt{n}} \left(\frac{4\delta_1 \ln n}{\pi\sqrt{n}} + \frac{(c/\sqrt{2})}{\sqrt{r^2-(c/\sqrt{2})^2}}\frac{(r+\sqrt{n})^2}{\pi n}\right. \\ &\qquad\qquad\left. + \frac{(c/\sqrt{2})}{\sqrt{r^2-(c/\sqrt{2})^2}}\frac{(\sqrt{n}-r)^2}{\pi n}\right)\frac{2r}{n} dr \\ &\leq \frac{(c/\sqrt{2}+w_n)^2}{n} + \frac{4\delta_1 \ln n}{\pi\sqrt{n}} + \frac{4c}{\pi\sqrt{n}} \leq \left(\frac{4\delta_1^2 \ln n}{\sqrt{n}} + \frac{4\delta_1}{\pi} + \frac{4c}{\pi \ln n}\right)\frac{\ln n}{\sqrt{n}} \\ &\leq \mu\frac{\ln n}{\sqrt{n}} \triangleq p_n\end{aligned}$$

where $\mu = (2 + 4\delta_1/\pi)$. □

Choose $\delta = 27\mu\pi$ and note that $a = \delta\sqrt{n}\ln n = 27\pi n p_n$. Substituting the uniform bound for $p_{i,j}$, obtained in Lemma 14, in (2.25)

$$\begin{aligned}
\Pr(L_i > \delta\sqrt{n}\ln n) &\leq \min_{s>0} e^{-s27\pi n p_n} \sum_{k=0}^{\infty} \Pr(N_n = k)[p_n(e^{9s} - 1) + 1]^k \\
&= \min_{s>0} e^{-s27\pi n p_n} \, \mathsf{E}[e^{N_n \ln(p_n(e^{9s}-1)+1)}] = \min_{s>0} e^{-s27\pi n p_n} e^{\pi n p_n (e^{9s}-1)} \\
&\leq e^{-\pi n p_n (3-e)},
\end{aligned}$$

where the last equality uses the known form of the moment generating function of a Poisson random variable and the last inequality uses $s = 1/9$. Then, from (2.24)

$$\begin{aligned}
\Pr(L(n) \leq \delta\sqrt{n}\ln n) &\geq 1 - \sum_{i=1}^{M_n} e^{-\pi n p_n (3-e)} \\
&= 1 - \exp\left(-\pi n p_n (3-e) + \ln M_n\right) \\
&\geq 1 - \exp\left(-\pi n \frac{\mu \ln n}{\sqrt{n}}(3-e) + \ln \frac{9n}{c^2}\right) \\
&\to 1 - 0 = 1 \text{ as } n \to \infty \ ,
\end{aligned}$$

which concludes the proof of Lemma 11. □

References

[1] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. on Inform. Theory*, vol. 46, no. 2, pp. 388 – 404, Mar 2000.

[2] S. Toumpis and A. J. Goldsmith, "Capacity regions for wireless ad hoc networks," *IEEE Trans. on Wireless Comm.*, vol. 2, no. 4, pp. 736 – 748, Jul 2003.

[3] L. Xie, L. and P. R. Kumar, "A network information theory for wireless communication: scaling laws and optimal operation," *IEEE Trans. on Inform. Theory*, vol. 50, no. 5, pp. 748 – 767, May 2004.

[4] F. Xue and P. R. Kumar, "The number of neighbors needed for connectivity of wireless networks," *Journal on Wireless Networking*, vol. 10, pp. 169–181, Mar 2004. [Online]. Available: http://dx.doi.org/10.1023/B:WINE.0000013081.09837.c0

[5] O. Dousse and P. Thiran, "Connectivity vs capacity in dense ad hoc networks," in *INFOCOM*, vol. 1, Mar 2004.

[6] O. Leveque and I. E. Telatar, "Information-theoretic upper bounds on the capacity of large extended ad hoc wireless networks," *IEEE Trans. on Inform. Theory,*, vol. 51, no. 3, pp. 858 – 865, Mar 2005.

[7] M. Franceschetti, O. Dousse, D. N. C. Tse, and P. Thiran, "Closing the gap in the capacity of wireless networks via percolation theory," *IEEE Trans. on Inform. Theory*, vol. 53, no. 3, pp. 1009 –1018, Mar 2007.

[8] A. Ozgur, O. Leveque, and D. N. C. Tse, "Hierarchical cooperation achieves optimal capacity scaling in ad hoc networks," *IEEE Trans. on Inform. Theory*, vol. 53, no. 10, pp. 3549 – 3572, Oct 2007.

[9] M. Grossglauser and D. N. C. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Trans. on Networking*, vol. 10, no. 4, pp. 477 – 486, Aug 2002.

[10] N. Bansal and Z. Liu, "Capacity, delay and mobility in wireless ad-hoc networks," in *INFOCOM*, vol. 2, Mar 2003, pp. 1553 – 1563.

[11] A. El Gamal, J. Mammen, B. Prabhakar, and D. Shah, "Optimal throughput-delay scaling in wireless networks: part i: the fluid model," *IEEE/ACM Trans. Netw.*, vol. 14, pp. 2568–2592, Jun 2006. [Online]. Available: http://dx.doi.org/10.1109/TIT.2006.874379

[12] R. Ramanathan, "On the performance of ad hoc networks with beamforming antennas," in *MobiHoc*, ser. MobiHoc '01. New York, NY, USA: ACM, 2001, pp. 95–105. [Online]. Available: http://doi.acm.org/10.1145/501426.501430

[13] A. Agarwal and P. R. Kumar, "Capacity bounds for ad hoc and hybrid wireless networks," *SIGCOMM*, vol. 34, pp. 71–81, July 2004. [Online]. Available: http://doi.acm.org/10.1145/1031134.1031143

[14] U. Kozat and L. Tassiulas, "Throughput capacity of random ad hoc networks with infrastructure support," in *MobiCom*, ser. MobiCom '03. New York, NY, USA: ACM, 2003, pp. 55–65. [Online]. Available: http://doi.acm.org/10.1145/938985.938992

[15] P. Li and Y. Fang, "Impacts of topology and traffic pattern on capacity of hybrid wireless networks," *IEEE Trans. on Mobile Computing*, vol. 8, no. 12, pp. 1585 – 1595, Dec 2009.

[16] Y. Dodge, *The Oxford Dictionary of Statistical Terms.* Oxford University Press, 2003.

[17] O. Dousse, F. Baccelli, and P. Thiran, "Impact of interferences on connectivity in ad hoc networks," *IEEE/ACM Trans. Netw.*, vol. 13, pp. 425–436, Apr 2005. [Online]. Available: http://dx.doi.org/10.1109/TNET.2005.845546

[18] O. Arpacioglu and J. Haas, Z., "On the scalability and capacity of wireless networks with omnidirectional antennas," in *IPSN*, 2004, pp. 169–177.

[19] D. Marco, E. Duarte-Melo, M. Liu, and D. L. Neuhoff, "On the many-to-one transport capacity of a dense wireless sensor network and the compressibility of its data," in *IPSN*, ser. IPSN'03. Berlin, Heidelberg: Springer-Verlag, 2003, pp. 1–16. [Online]. Available: http://portal.acm.org/citation.cfm?id=1765991.1765993

[20] S. R. Kulkarni and P. Viswanath, "A deterministic approach to throughput scaling in wireless networks," *IEEE Trans. on Inform. Theory*, vol. 50, no. 6, pp. 1041 – 1049, Jun 2004.

[21] E. Duarte-Melo, A. Josan, M. Liu, D. L. Neuhoff, and S. S. Pradhan, "The effect of node density and propagation model on throughput scaling of wireless networks," in *ISIT*, Jul 2006, pp. 1693 –1697.

[22] G. Grimmett, *Percolation*, 2nd ed., ser. Grundl. Math. Wissen. New York: Springer-Verlag, 1999, no. 321.

[23] H. Kesten, *Percolation Theory for Mathematicians.* Boston, MA: Birkhuser,, 1982.

CHAPTER III

Reliability Efficiency Tradeoff in Wireless Sensor Networks

3.1 Introduction

Distributed source coding allows a number of spatially separated correlated sources to encode in a way that exploits their mutual correlations. In particular, as illustrated in Fig. 3.1, suppose N identical, discrete-time, discrete-valued, jointly stationary and ergodic sources, denoted X_1,..., X_N, are to be independently encoded for joint decoding. Then Slepian-Wolf theory [1, 2, 3] shows that encoders for the N sources can be designed to encode temporal blocks at a rate approximately equal to $\frac{1}{N}H_\infty(X_1,..., X_N)$ bits per source per time unit (averaged over the N sources), and a joint decoder can reconstruct X_1,..., X_N with arbitrarily small error probability, where H_∞ denotes entropy-rate. If the correlation between sources is significant, this rate can be substantially smaller than $\frac{1}{N}\sum_{i=1}^{N} H_\infty(X_i)$, the minimum encoding rate if each source is optimally encoded in a way that it can be decoded without access to the encodings of the other sources. This, for example, could be very useful in a field gathering sensor network in which each sensor measures a quantity like temperature that varies across the field and that is correlated with the measurements made at nearby sensors.

Although the encoders operate separately, in its basic form, the decoding is done

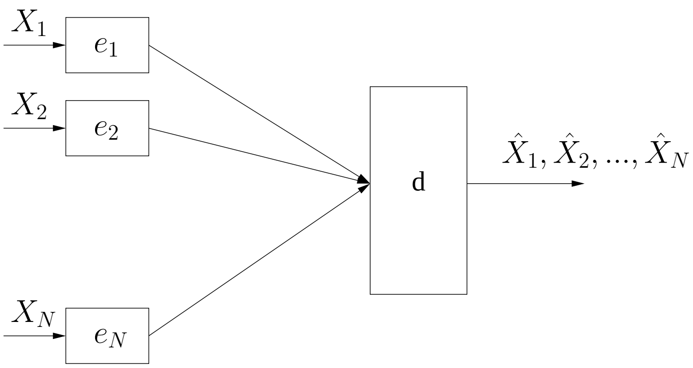


Figure 3.1: Distributed coding system. e_1,..., e_N do independent encoding while decoder d does joint decoding.

jointly with the result that if one encoder output is missing, for example, if one encoder fails, as might happen in a sensor network when a sensor runs out of energy, then the decoder can fail to correctly decode any of the sources. Thus a system that employs distributed encoding to encode a large number of sources could be unreliable due to its high sensitivity to the loss of encoded data from just one source.

One might, of course, limit the sensitivity to encoder failure by grouping sources into blocks of say K sources and encoding/decoding each group separately from the others. In this case, the failure of one encoder causes the failure of decoding of itself and of all other sources in the group, but not of the sources in other groups. However, this reduces the collaborative gain. To simplify the discussion and for concreteness, from now on we assume that the N sources X_1,..., X_N come from a spatio-temporal collection $\{X_{ij} : -\infty < i, j < \infty\}$, where i indicates the spatial identity of the source and j indicates time, such that (a) each source $X_i = \{X_{ij} : -\infty < j < \infty\}$ is temporally IID and identical to every other source, (b) the spatial sample functions at different times are independent, i.e. ..., $X^{(-1)}, X^{(0)}, X^{(1)}$,... are independent, where $X^{(j)} = \{X_{ij} : -\infty < i < \infty\}$ is the sample function at time j, and (c) the sources $\{..., X_{-1}, X_0, X_1, ...\}$ form a spatially stationary collection[1]. Among other things, these assumptions imply that $H_\infty(X_1,..., X_N)$ equals the joint entropy $H_1(X_1,..., X_N)$ of

[1] Actually, (b) and (c) imply (a), but the latter is stated for transparency

the collection of time-one random variables from each source, and that $H_\infty(X_i)$ is the same for all i and equals $H_1(X_1)$, the entropy of the first random variable from the first source. It is also well known that $\frac{1}{K}H_1(X_1,...,X_K)$ decreases with increasing K (strictly so, if the sources are correlated). Therefore, as K decreases, the encoding rate of the aforementioned *grouping* scheme increases from $\frac{1}{N}H_1(X_1,...,X_N)$ to $\frac{1}{K}H_1(X_1,...,X_K)$. Thus, a trade-off is possible: increased reliability (reduced sensitivity) for a loss in encoding efficiency, with smaller values of K causing a larger trade-off.

Another scheme for limiting sensitivity is to encode each source X_i at rate of $H_1(X_i|X_1, ..., X_{i-1})$, in which case the average rate across all sources will again be $\frac{1}{N}H_1(X_1,...,X_N)$, the minimal value. Slepian-Wolf theory [1, 3] shows that if the data from all the encodings at these rates is received by the decoder, it can reconstruct $X_1,...,X_N$ with arbitrarily high probability. Moreover, it is natural to expect that if the data from only the ith encoder is lost, a decoder can still correctly decode with high probability[2] the data from sources $X_1,...,X_{i-1}$. For example, if data from encoders is lost independently with probability p, then, as we compute later, this implies that on the average $1-\frac{1}{N}\frac{1-p}{p}[1-(1-p)^N]$ fraction of the sources cannot be decoded. While this *sequential* scheme gains reliability without a loss of coding efficiency, when N is large, the fraction of sources that are not decodable approaches one, meaning this scheme is not so useful. On the other hand, when N is large, this scheme can be combined to good effect with the grouping scheme of the previous paragraph.

The goal of this chapter is to propose, quantitatively analyze and compare schemes, such as those in the previous two paragraphs, for trading coding efficiency for increased reliability (reduced sensitivity). A long term goal is to understand the fundamental limits of this trade-off. While this chapter does not find the ultimate limits,

[2] That this is indeed possible is shown in Section 3.4.

its results do provide bounds as well as practical schemes for attaining increased reliability.

We first explore schemes we call *rigid*, including the two schemes above, in which successful decoding is presumed to occur for source i if and only if all encodings from a designated set of sources S_i, called the *dependence set* for i, is received. For example, in the case of the grouping scheme above, S_i consists of all sources in the same group as i, while for the sequential scheme, $S_i = \{1,...,i\}$. Next, exploiting the fact that Slepian-Wolf theory uses binning-type encoding, and that binning-type encoding does not actually rely on knowledge of a specific dependence set, we consider *flexible* schemes in which decoding is presumed to occur for source i if and only if all sources in at least one of a collection of dependence sets is received. Rigid is a special case of flexible, and as we shall see, flexible schemes perform better.

In order to be able to quantify the reliability/sensitivity of a given scheme we assume that encoders fail independently with probability p, and that once they fail, they are lost forever. We make the latter assumption because the encoding methods to be considered operate on temporal blocks of data, and in this initial effort we do not wish to deal with the possibility of an encoder failing during a block, or of the probability of encoder failure during a block increasing as the block length increases, or of having to make the unrealistic assumption that block loss probabilities are independent of block length. As one example, this scenario models the situation that sensors are dropped from an airplane and fraction p fail upon hitting the ground. It also applies to the situation that all encoders are functioning initially; however, we know that encoders will fail over time, so we design the system for a specified nonzero failure probability p.

Continuing with the goal of quantifying reliability/sensitivity, we adopt two primary performance measures: *rate* R and *loss factor* L. By *rate* we mean the average number of encoded bits per time unit per source. By *loss factor* we mean the expected fraction of sources that are decoded incorrectly. We focus mainly on schemes we call *lossless*, in which incorrect decoding is due primarily to failure of one or more encoders. In other words, in the absence of encoder failure, the coding is essentially lossless[3].

Let F denote a random variable representing the fraction of sources that are incorrectly decoded. Then aside from loss factor L, which is the expected value of F, one might also be interested in the variance of F, or some other statistic such as the cumulative distribution function of F. Though in this initial study we focus mainly on the loss factor L, we also give a few results for the variance of F. When the number of sources N is much larger than the group size K, the law of large numbers implies that the probability distribution of F concentrates near $L = E[F]$. The variance estimates will confirm this. Thus, the loss factor L is a generally good measure of reliability.

We should make clear that the rigid schemes in this chapter were first proposed by Marco and Neuhoff [4]. However, this chapter is the first to provide a rigorous theoretical formulation for the reliability-efficiency problem. This chapter also extends the Slepian-Wolf theorem that is required to evaluate the rates for the rigid schemes. We also provide a more comprehensive analysis of these schemes in terms of the variance of the loss factor of the schemes. As such, we do keep the introduction of the schemes in this chapter.

The chapter is organized as follows. Section 3.2 reviews distributed coding and the Slepian-Wolf theorem. Section 3.3 introduces rigid systems, the model for encoder failure and the two performance measures. Section 3.4 extends the Slepian-Wolf Theorem to the situation that not all encoder outputs are available at the decoder. Several rigid schemes are proposed and analyzed in Section 3.5, and their performances compared in Section 3.6. To make the analysis and comparison, we consider

[3] Like codes in the usual Slepian-Wolf theory, in the absence of encoder failure these codes correctly decode with probability less than, but very close to, one.

a scenario in which the N sources form a Markov chain of IID sources. The specific parameters of the sources come from considering uniform quantization of samples of an underlying Gauss-Markov random process. Section 3.7 extends the formulation to include flexible schemes. Several such schemes are proposed and analyzed in Section 3.8, and the performances of flexible and rigid schemes are compared in Section 3.9, where it is found that flexible schemes significantly outperform rigid, albeit with some increase in complexity. Appendix A proves the extended Slepian-Wolf lemma of Section 3.4, and Appendix B contains the loss factor calculations for the coding schemes introduced in the chapter.

Finally, we mention the related work of which we are aware. Coleri and Varaiya [5] suggest an explicit multipath communication scheme in which a source s_1 compresses its data with respect to a second source s_2 only if s_1 acts as a relay for s_2. This ensures that if data from s_1 gets to the decoder the side information required to decode it from s_2 gets to the decoder as well. Chen and Berger [6] consider the lossy version of a similar problem, where they study the trade-off between system robustness and compression efficiency. A coding akin to flexible coding has also been used in construction of codes for the multiple description problem [7]. Flexible coding also has similarities with the CEO problem [8], where N sources independently observe and encode corrupted versions of a single source and communicate it to a central decoder which tries to reproduce the source within some desired fidelity. Applications of flexible coding can be found in video coding literature. For example in [9], depending on playback order, future or past frames are available at the decoder. An intermediate frame that is flexibly encoded can be decoded using either of those frames.

3.2 Slepian-Wolf Distributed Lossless Coding

In this section we review the problem of distributed lossless coding, first investigated by Slepian and Wolf [1, 2, 3]. First consider the encoding of two correlated

IID sources X and Y. From now on we drop subscripts and let H, rather than H_1 or H_∞, denote entropy or entropy-rate. From point-to-point information theory, we know that in order to encode a source X losslessly, a rate of $R > H(X)$ is necessary and sufficient. Thus, if there are two sources $(X, Y) \sim P_{X,Y}(x, y)$, then to encode and decode each of them individually would require rates $R_X > H(X)$ and $R_Y > H(Y)$. However, if we do joint encoding and decoding, the point-to-point formula would suggest a rate of $R = R_X + R_Y > H(X, Y)$ to be sufficient. The question arises, what rate would be sufficient in case the encoding is done independently and decoding is performed jointly. The surprising answer, given by Slepian and Wolf, stated that a sum rate of $H(X, Y)$ is also sufficient in the case of separate encoding and joint decoding.

More generally, we consider a system of N sources $X_1,..., X_N$, as described in the introduction, which we think of as spatially separated, each having an encoder that sends the encoded source descriptions to a joint decoder. Let $\mathcal{X}$ denote the common alphabet of the random variables in each source, and let $\mathcal{X}^m$ denote the set of all m-tuples from $\mathcal{X}$. Let $X_i^m = (X_{i1},..., X_{im})$, and similarly define x_i^m.

Definition 15 *An $(R_1,..., R_N, m)$ distributed source code for $X_1,..., X_N$ consists of a set of N encoder mappings*

$$e_i : \mathcal{X}^m \to \{1,..., \lceil 2^{mR_i} \rceil\}$$

and a decoder mapping

$$d : \times_{i=1}^{N} \{1,..., \lceil 2^{mR_i} \rceil\} \to \mathcal{X}^{mN}\,.$$

A system with the above code operates as follows. For each i, the encoder for the ith source takes a temporal block of m source symbols x_i^m and produces a sequence of bits $b_i = e_i(x_i^m)$. The decoder takes as its input the output bits from all the encoders and produces N blocks as its output $\underline{\hat{x}} = d(b_1,...b_N)$, where $\underline{\hat{x}} = (\hat{x}_1^m,..., \hat{x}_N^m)$. We consider R_i to be the *rate* of the ith encoder.

Definition 16 *The probability of error, P_e, of a distributed code is*

$$P_e = P_e(e_1,..., e_N, d) = \Pr\left(d(e_1(X_1^m),..., e_m(X_N^m)) \neq (X_1^m,..., X_N^m)\right) .$$

In the following and elsewhere we use the notation $a_k \overset{k\to\infty}{\leq} b$ to mean that a_k is a sequence such that $\limsup_{k\to\infty} a_k \leq b$. Moreover, if $\underline{a_k} = (a_{k,1},..., a_{k,n})$ is a sequence of vectors and $b = (b_1,..., b_N)$, then $\underline{a_k} \overset{k\to\infty}{\leq} \underline{b}$ means $\limsup_{k\to\infty} a_{k,i} \leq b_i$ for $i = 1$ to n.

Definition 17 *A rate vector $\underline{r} = (r_1,..., r_N)$ is* achievable for $(X_1,..., X_N)$ *if there exists a sequence of distributed codes with rates $\underline{R_k} = (R_{k,1},..., R_{k,N})$ and error probabilities $P_{e,k}$ such that $\underline{R_k} \overset{k\to\infty}{\leq} \underline{r}$ and $P_{e,k} \to 0$ as $k \to \infty$. The* Slepian-Wolf achievable rate region, *denoted $\mathcal{R}_{SW}$, is the set of all achievable rate vectors for $(X_1,..., X_N)$.*

Slepian and Wolf [1, 3] gave the following characterization of the achievable rate region.

Theorem 18 *A rate vector $(r_1,..., r_N)$ is achievable for $(X_1,..., X_N)$ by distributed source coding if and only if*

$$r_T \geq H(X_T|X_{T^c})$$

for all $T \subset \{1,..., N\}$, where $r_T = \sum_{i\in T} r_i$, $X_T = \{X_i : i \in T\}$, and H denotes entropy. Equivalently, $\mathcal{R}_{SW}$ is the set of all rate vectors satisfying the above.

For example, one can straightforwardly use this theorem to show that the rate vector $\underline{r}$ is achievable if $r_i = H(X_i|X_1,..., X_{i-1})$, $i = 1,..., N$. Moreover, if in addition to stationarity $X_1,...,X_N$ forms a stationary Markov chain, then $r_i = \frac{1}{N}H(X_1,...,X_N)$, $i = 1,...,N$ is achievable.

3.3 Rigid Codes

In the previous section, we looked at distributed coding for N sources, when all the encoder outputs are present at the decoder. In this section, we propose a system model that takes into account encoder failures. For this purpose, we again consider a collection of sources $X_1,...,X_N$, as described in the introduction, for which we again consider an $(R_1,...,R_N,m)$ distributed source coding system with N encoders with block length m, $e_1,...,e_N$, one for each source. However, we will now think of a decoder as being composed of N (sub)decoders, $d_1,...,d_N$, one for each source. Moreover, the domain and range of these decoders must be expanded to include the possibility that one or more of the encoder outputs is not available, and that the decoder might not attempt to decode. Specifically, the decoder for the ith source is a mapping

$$d_i : \times_{j=1}^{N}\{1,...,\lceil 2^{mR_j}\rceil, \phi_j\} \to \mathcal{X}^m \cup \{\Phi_i\}$$

where ϕ_j is a symbol indicating that no encoding of source j is received, and Φ_i is a symbol indicating that decoding of source i is attempted and an error is declared for this source. The ith decoder operates by applying d_i with arguments consisting of whatever encoder outputs are received and ϕ_j's where no encoding is received. To formalize this, let I_i be the indicator function representing the success or failure of encoder i. That is, $I_i = 0$ if encoder i fails, and $I_i = 1$ otherwise. Also, let

$$\hat{b}_i = \begin{cases} b_i = e_i(x_i^m), & I_i = 1 \\ \phi_i, & \text{otherwise} \end{cases}.$$

Then it is $\hat{b} = (\hat{b}_1,..., \hat{b}_N)$ to which the decoders are applied. In particular, the ith decoder produces

$$\hat{x}_i = d_i(\hat{b}_1,..., \hat{b}_N)\,.$$

As mentioned in the introduction, we assume that encoders fail independently of each other with probability p, which we will refer to as the *encoder failure probability*[4]. We assume that the encoders, once they fail remain dead for all time. Note that under our assumption, $I_1,..., I_N$ are i.i.d. with $\Pr(I_i = 0) = p$.

As also mentioned in the introduction, in order to decrease sensitivity to encoder failures, instead of requiring bits from all the encoders, the decoder for source i may only need bits from a subset $S_i \subset \{1,..., N\}$ of the encoders. We call S_i the *dependence set* for source i and $\underline{S} = (S_1,..., S_N)$ the *dependence structure*. Since the decoder for source i requires at least some bits from the encoding of i, we make the assumption that $i \in S_i$ for all i. To capture the idea of decoding from a subset of encoders, we say a code *conforms* to a dependence structure $\underline{S}$, if there exist functions $\tilde{d}_{i,S_i} : \times_{j\in S_i}\{1,...\lceil 2^{mR_j}\rceil\} \to \mathcal{X}^m$, $i = 1,..., N$, such that

$$d_i(\hat{b}_1,..., \widehat{b}_N) = \begin{cases} \tilde{d}_{i,S_i}(\hat{b}_{S_i}), & \text{if } I_j = 1 \text{ for all } j \in S_i \\ \Phi_i, & \text{else} \end{cases} \tag{3.1}$$

where $\hat{b}_{S_i} = \{\hat{b}_j : j \in S_i\}$.

We now define a *rigid distributed coding system* s for $X_1,..., X_N$ to be a dependence structure $\underline{S} = (S_1,..., S_N)$ and an $(R_1,..., R_N, m)$ code conforming to $\underline{S}$.

The dependence structure $\underline{S}$ of a rigid system s can be visualized in a natural way as a balanced bi-partite graph $D = (V_1, V_2, E)$, where $V_1 = \{1,..., N\}$ are graph

[4] Though we refer to *encoder failure*, there could actually be some other mechanism that prevents the decoder from receiving the encoding of a source.

nodes corresponding to the encoders, $V_2 = \{1,..., N\}$ are nodes (with the same names) corresponding to the decoders, and E is the set of edges, where if $j \in S_i$, there is a directed edge (j, i) from $j \in V_1$ to $i \in V_2$. Note that since dependence graphs are complete representations for dependence structures, we can use the two terms interchangeably.

3.3.1 Performance

The metrics used to measure the performance of a system are average rate and loss factor.

Definition 19 *The* average rate *of a rigid system s, denoted $R_{av}(s)$, is the average of the rates at which the N sources are encoded:*

$$R_{av}(s) = \frac{1}{N}\sum_{i=1}^{N} R_i \,, \tag{3.2}$$

where R_i is the rate of the ith encoder.

Definition 20 *The* loss factor *of a rigid system s when the encoder failure probability is p, denoted $L(s, p)$, is the expected fraction of source values that cannot be correctly reconstructed at the decoder, i.e.,*

$$\begin{aligned} L(s,p) &= E\left[\frac{1}{N}\sum_{i=1}^{N} I\left(X_i^m \neq d_i(\hat{b}_1,..., \hat{b}_N)\right)\right] \\ &= \frac{1}{N}\sum_{i=1}^{N} \Pr\left(X_i^m \neq d_i(\hat{b}_1,..., \hat{b}_N)\right), \end{aligned} \tag{3.3}$$

where $I(A)$ is an indicator function equaling 1 if the event A occurs, and 0 otherwise.

The loss factor of a rigid system is influenced by two phenomena: encoder failures and coding errors. The latter occurs when the decoder fails to decode correctly even when bits from all the encoders are received. To quantify the contributions to loss factor from these two phenomena, let A_i denote the event that $I_j = 1$ for all $j \in S_i$, and let $P_{e,i}$ denote the probability of erroneously decoding source i given A_i. Then

$$L(s,p) = \frac{1}{N}\sum_{i=1}^{N} P_{e,i}\,\Pr(A_i) + \frac{1}{N}\sum_{i=1}^{N}\Pr(A_i^c)\,, \tag{3.4}$$

where

$$\Pr(A_i) = (1-p)^{|S_i|}\,. \tag{3.5}$$

We see from (3.4) that the loss factor of a system decomposes into two terms: the first depending on the coding error and dependence structure, and the second depending only on the dependence structure.

The first term can be upper bounded as

$$\frac{1}{N}\sum_{i=1}^{N} P_{e,i}\,\Pr(A_i) \le \frac{1}{N}\sum_{i=1}^{N} P_{e,i} = L(s,0)\,.$$

We call $L(s,0)$ the *code loss factor*, as here the entire probability of error comes from coding errors. Since the second term in (3.4) depends only on the dependence structure $\underline{S}$, we give it a corresponding name.

Definition 21 *The* dependence structure loss factor[5] *is*

$$L_d(\underline{S},p) = 1 - \frac{1}{N}\sum_{i=1}^{N}\Pr(A_i)\,. \tag{3.6}$$

Combining (3.4) and the definitions of $L(s,0)$ and $L_d(\underline{S},p)$, it is easily seen that

$$L_d(\underline{S},p) \le L(s,p) \le L_d(\underline{S},p) + L(s,0)\,. \tag{3.7}$$

[5] It will sometimes be referred to as the *dependence graph loss factor*.

Also, note that $L_d(\underline{S}, p)$ can never be less than p, due to the convention in (3.1) that decoder i will not attempt to decode whenever encoder i fails, which happens with probability p. In the case of lossless codes, $L(s, 0)$ is small and hence the loss factor of a system is almost entirely determined by the dependence structure. For this reason, from now on we primarily consider $L_d(\underline{S}, p)$ to be the loss factor. Also, when designing a rigid system we think mainly in terms of choosing a dependence structure.

3.3.2 Achievable rates and the rate-loss region and function

With the goal of focusing mainly on dependence structure, we now consider the concept of *rate* for a dependence structure. Since in this chapter we focus on lossless codes, i.e. codes that are lossless in the absence of encoder failures, the following definitions include the constraint that $L(s, 0)$ be asymptotically small.

Definition 22 *A rate vector* $\underline{r} = (r_1, ..., r_N)$ *is said to be* achievable for dependence structure $\underline{S} = \{S_1, ..., S_N\}$ *if there exists a sequence of rigid systems,* s_k*, each with structure* $\underline{S}$*, having rate vectors* $\underline{R}_k = (R_{k,1}, ..., R_{k,N})$ *such that* $\underline{R}_k \overset{k\to\infty}{\leq} \underline{r}$*, and* $L(s_k, 0) \overset{k\to\infty}{\longrightarrow} 0$*. Let* $\mathcal{R}(\underline{S})$ *be the set of achievable rate-vectors for* $\underline{S}$*, and let* $R_{min}(\underline{S}) = \inf\{\frac{1}{N}\sum_{i=1}^{N} r_i : \underline{r} \in \mathcal{R}(\underline{S})\}$ *denote the* minimum rate *for* $\underline{S}$*.*

Clearly, $\mathcal{R}(\underline{S}) \subset \mathcal{R}_{SW}$ for every $\underline{S}$, and $\mathcal{R}(\underline{S}) = \mathcal{R}_{SW}$ when $S_i = \{1, ..., N\}$ for each i. Csiszar and Korner [10] have found a single-letter characterization of $\mathcal{R}(\underline{S})$. Since it involves auxiliary random variables, it can be difficult to evaluate.

Definition 23 *A rate-loss pair* (r, l) *is said to be* achievable *for source* $\underline{X} = (X_1, ..., X_N)$*, dependence structure* $\underline{S} = (S_1, ..., S_N)$*, and encoder loss probability* p *if there exists a sequence of rigid systems* s_k*, each conforming to* $\underline{S}$*, such that* $R_{av}(s_k) \overset{k\to\infty}{\leq} r$*,* $L(s_k, p) \overset{k\to\infty}{\leq} l$*, and* $L(s_k, 0) \overset{k\to\infty}{\longrightarrow} 0$*. The* achievable rate-loss region *for* $\underline{X}$*,* $\underline{S}$ *and*

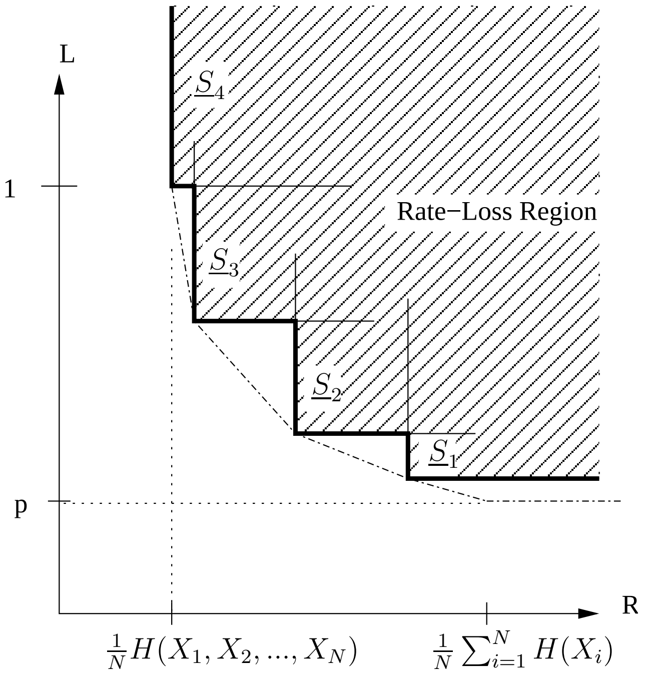


Figure 3.2: A schematic rate-loss region $\mathcal{RL}$. Each quadrant corresponds to a specific dependence structure $\underline{S}$. The loss-rate function $\mathcal{L}(r)$ is shown by the thick line. The convex hull is shown by the dash-dot line.

p, denoted $\mathcal{RL}(\underline{S})$*, is the set of all achievable rate-loss pairs. The* overall achievable rate-loss region *for this source and encoder loss probability*[6] *is* $\mathcal{RL} = \cup_{\underline{S}}\mathcal{RL}(\underline{S})$.

The following lemma, proved in Appendix 3.A, shows that the rate-loss region $\mathcal{RL}(\underline{S})$ has a simple form.

Lemma 24 *The rate-loss region* $\mathcal{RL}(\underline{S})$ *for a dependence structure* $\underline{S}$ *is the north-east quadrant* $\{(r, l) : r \geq R_{min}(\underline{S}), l \geq L_d(\underline{S}, p)\}$.

Definition 25 *The* loss-rate function*,* $\mathcal{L}(r)$*, is the minimum loss factor of all systems with rate less than or equal to r, i.e.,*

$$\mathcal{L}(r) = \inf\{l : (r, l) \in \mathcal{RL}\}\,.$$

[6] Note that $\mathcal{RL}(\underline{S})$ and $\mathcal{RL}$ depend implicitly on the source and encoder loss probability.

Fig. 3.2 shows a schematic of an achievable rate-loss region and the corresponding loss-rate function. Since the number of distinct dependence structures is finite, the $\mathcal{RL}$ region has a staircase form. If we were to allow time-sharing of systems with different dependence structures, then the lower boundary of the $\mathcal{RL}$ region would become the convex hull of the corner points of all $\mathcal{RL}(\underline{S})$ regions.

3.4 Extended Slepian-Wolf Theorem

Theorem 18 gives the achievable rate region when all encoder descriptions are present at the decoder. However, we need the achievable rate region such that source i can be decoded from encoder descriptions in (only) S_i. Theorem 27 below, an extension of Theorem 18 that is proved in Appendix 3.B, gives sufficient conditions on encoder rates to permit this.

Definition 26 *A pair (i, S) consisting of source i and its dependence set S is* supported by *a rate vector $\underline{r}$ if*

$$r_U \geq H(X_U | X_{S-U})$$

for all subsets U of S that contain i.

Theorem 27 (Extended Slepian-Wolf) *Given $\underline{r} = (r_1, ..., r_N)$, there exists a sequence of sets of encoder mapping $(e_{k,1}, ..., e_{k,N})$, $k = 1, 2, ...$, with blocklengths denoted m_k, and rates $\underline{R}_k \overset{k\to\infty}{\leq} \underline{r}$ such that for each (i, S) supported by $\underline{r}$ there exists a corresponding sequence of decoders $\tilde{d}_{k,i,S}$ that can decode source i from (only) the encoded descriptions of X_S with probability of error converging to zero, i.e.,*

$$P_e(e_{k,S}, \tilde{d}_{k,i,S}) \triangleq \Pr(X_i^{m_k} \neq \tilde{d}_{k,i,S}(e_{k,S}(X_S^{m_k}))) \to 0 \text{ as } k \to \infty$$

where $e_{k,S}(X_S^{m_k}) = \{e_{k,j}(X_j^{m_k}) : j \in S\}$.

Note that since there are only finitely many possibilities for (i, S), the convergence of $P_e(e_{k,S}, \tilde{d}_{k,i,S})$ shown in this theorem is uniform over (i, S).

Corollary 28 *In the setting of rigid codes, the rate vector $\underline{r} = (r_1,..., r_N)$ is achievable for dependence structure $\underline{S} = (S_1,..., S_N)$ if S_i is supported by $\underline{r}$ for each $i \in \{1,..., N\}$.*

Theorem 27 contains the original Slepian-Wolf result as a special case, namely, $S_i = \{1, 2,..., N\}$ for all i. It also shows that the design of the encoders need not depend on what i and S_i we may have in mind. That is, it establishes the existence of encoders that are universal in the sense that their design is independent of the dependence structure of the decoders.

As an application of the theorem, consider the sequential encoding scheme in the introduction where, for $i = 1,..., N$, the encoder for source i transmits at rate $r_i = H(X_i|X_1,..., X_{i-1})$. Since these rates satisfy the Slepian-Wolf conditions, the original Slepian-Wolf theorem shows that if bits from all N sources are received at the decoder, all of them can be decoded correctly. Now, however, one can verify that for each i the rate vector $\underline{r}$ supports $(i, \{1,..., i\})$, and consequently the above theorem verifies that there exist encoders with these rates such that for each i there is a decoder that can correctly decode sources 1 through i with high probability whenever the encodings of the same sources are received, regardless of whether the encodings of sources $i + 1$ through N are received or not.

Definition 29 *The* extended Slepian-Wolf rate region *for a dependence structure $\underline{S}$, denoted $\mathcal{R}_{SW}(\underline{S})$, is the set of all rate vectors satisfying the condition of the corollary.*

Clearly, $\mathcal{R}_{SW}(\underline{S}) \subset \mathcal{R}(\underline{S})$. Moreover, Korner and Martin have found examples where the containment is strict. It follows from this and the definition of $R_{min}(\underline{S})$ that

$$R_{min}(\underline{S}) \leq \inf_{(r_1,..,r_N) \in \mathcal{R}_{SW}(\underline{S})} \frac{1}{N} \sum_{i=1}^{N} r_i$$

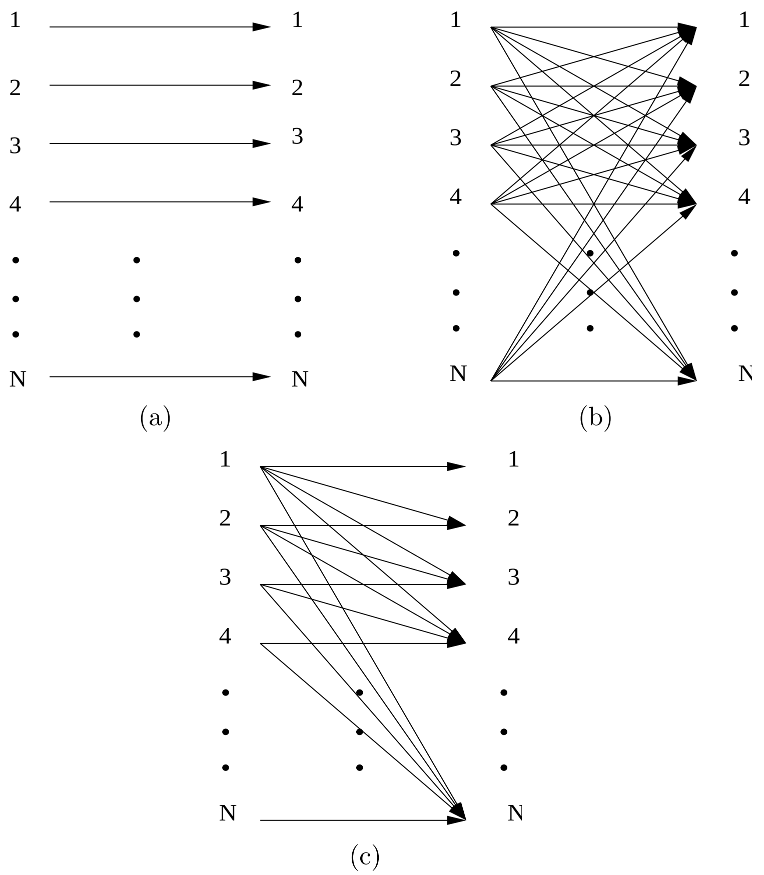


Figure 3.3: Dependence graphs (a) No Slepian-Wolf (b) Complete Slepian-Wolf (c) Sequential Slepian-Wolf

with strict inequality for some $\underline{S}$.

3.5 Several Rigid Schemes

In this section we explore several simple families of rigid encoding systems. We refer to such families as *schemes*. Among other things they induce upper bounds to the loss-rate function.

1) No Slepian-Wolf

Consider the dependence structures $\underline{S}$ whose graph is shown Fig. 3.3(a). For a system that conforms to $\mathcal{S}$, the decoder for a given source requires that the encoder for that source be active, i.e. it has not failed. Note that such a system can be thought of as N point-to-point coding systems, and hence the encoders do not need to perform any distributed coding. Hence we call this the No Slepian-Wolf scheme. For source i the minimum rate at which it can transmit is $H(X_i)$. Also, the loss factor for the graph can be calculated using (3.6). Thus, for the No Slepian-Wolf scheme the average rate and loss factor are

$$\begin{aligned} R_{min}(\underline{S}) &= \frac{1}{N}\sum_{i=1}^{N} H(X_i) \\ L_d(\underline{S}, p) &= p\,. \end{aligned}$$

This scheme has the lowest possible loss factor and the highest possible average rate.

2) Complete Slepian-Wolf

Consider the complete dependence structure $\underline{S}$ whose graph is shown in Fig. 3.3(b). As mentioned earlier, the original Slepian-Wolf Theorem (Thm. 18) shows that the achievable rates r_1,..., r_N can be chosen so that $\sum_{i=1}^{N} r_i = H(X_1,..., X_N)$. For example, by choosing $r_i = H(X_i|X_1,..., X_{i-1})$ for each i, or if X_1,..., X_N forms a Markov chain, by choosing $r_i = \frac{1}{N}H(X_1,..., X_N)$, for each i. The loss factor is also easily calculated, leading to

$$\begin{aligned} R_{min}(\underline{S}) &= \frac{1}{N}H(X_1,..., X_N) \\ L_d(\underline{S}, p) &= 1 - (1-p)^N\,. \end{aligned}$$

This scheme has the lowest possible average rate and the highest possible loss factor.

3) Sequential Slepian-Wolf

Consider the dependence structure $\underline{S}$ whose graph in Fig. 3.3(c), which describes the sequential scheme of the introduction. As mentioned earlier, rates $r_i = H(X_i|X_1,..., X_{i-1})$ are achievable. As with the previous example, they induce $R_{min}(\underline{S})$. The loss factor

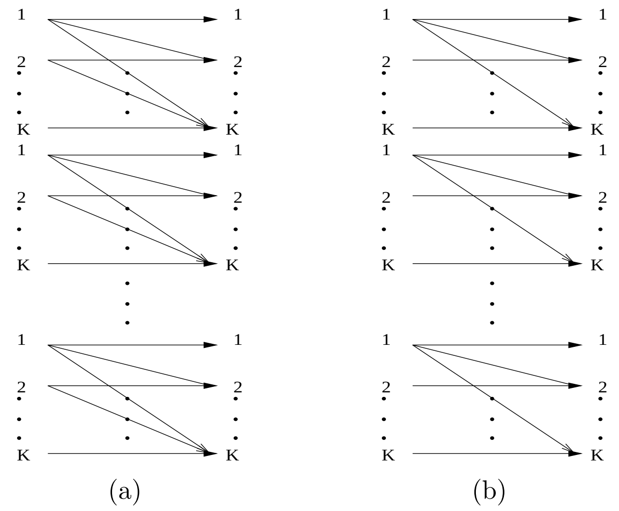


Figure 3.4: Dependence graphs. (a) Clustered Sequential Slepian-Wolf (b) Master-Slave, with source 1 as the master.

can again be calculated, leading to

$$R_{min}(\underline{S}) = \frac{1}{N}\sum_{i=1}^{N} H(X_i|X_1,\ldots,X_{i-1}) = \frac{1}{N}H(X_1,\ldots,X_N)$$

$$L_d(\underline{S},p) = 1 - \frac{1}{N}\sum_{i=1}^{N}(1-p)^i = 1 - \frac{1}{N}\frac{1-p}{p}[1-(1-p)^N]\,.$$

This scheme also has the lowest possible average rate, at a lower loss factor than Complete Slepian-Wolf. However, for p fixed, the loss factor goes to 1 as $N \to \infty$.

4) Clustered Sequential Slepian-Wolf

Consider the family of dependence structures $\underline{S}_K$, illustrated in Fig 3.4(a). In these structures, indexed by K, the N sources of both V_1 and V_2 are divided into clusters of size K. In each of these clusters, one performs sequential Slepian-Wolf coding. (Sequential Slepian-Wolf and No Slepian Wolf schemes are the special cases corresponding to $K = N$ and $K = 1$, respectively.) For systems conforming to this kind of dependence graph, the decoder works similarly to sequential Slepian-Wolf — in order to decode source i in a given cluster, the decoder needs to receive bits from encoders 1 through i in that cluster. By the stationarity assumption, the average rate and loss factor of the system are the same as for one cluster. It follows the average rate and loss factor are given by the formulas for Sequential Slepian-Wolf with N replaced by K. Thus,

$$R_{min}(\underline{S}_K) = \frac{1}{K}\sum_{i=1}^{K} H(X_i|X_1,..., X_{i-1}) = \frac{1}{K}H(X_1,...X_K)$$
$$L_d(\underline{S}_K, p) = 1 - \frac{1}{K}\frac{1-p}{p}[1-(1-p)^K] \, .$$

5) Master-Slave Scheme

Consider the family of dependence structures $\underline{S}_K$, indexed by K, whose graphs are illustrated in Fig 3.4(b). Here, N sources are divided into clusters of size K, and in each cluster, one source is designated as the *master* m. The dependence set for the master consists of itself only. All other sources in the cluster are designated *slaves*, with dependence set comprising itself and the master. Theorem 27 gives the minimal rates of $R_m = H(X_m)$ for the master and $R_i = H(X_i|X_m)$ for the slave. By stationarity, the average rate of the graph is same as the average rate of a cluster. Thus

$$R_{min}(\underline{S}_K) = \frac{1}{K}\left(H(X_m) + \sum_{j=1, j\neq m}^{K} H(X_j|X_m)\right) .$$

To minimize the average rate, the master should be positioned appropriately in the cluster. The loss factor of the graph is

$$L_d(\underline{S}_K, p) = 1 - \frac{1}{K}[(1-p) + (K-1)(1-p)^2] = p(2-p) - \frac{1}{K}p(1-p) \, .$$

3.6 Comparison of Rigid Schemes

In this section, we compute and compare the performance of the schemes introduced in the previous section for a representative example. To facilitate the computation of rates, we assume $X_1,...,X_N$ comes from a stationary Markov sequence. Among other things, this implies that $H(X_i|X_1,\ldots,X_{i-1}) = H(X_2|X_1)$ and $H(X_1,...,X_K) = H(X_1) + (K-1)H(X_2|X_1)$. To choose the parameters of this Markov process, we consider a one-dimensional sensor network that takes N samples of a zero-mean, unit variance, one-dimensional, continuous-space, stationary Gaussian random process, $Z(s)$, $-\infty < s < \infty$, with exponential auto-correlation $\rho_Z(\tau) = e^{-|\tau|}$. (Such a process is Markov.) The sensors are uniformly spaced over a unit interval at a distance of $1/N$. Each sensor quantizes its sample with an infinite-level uniform scalar quantizer with step-size $\Delta = 0.1$ and a level at the origin. The output of this quantizer is a discrete-time, amplitude-discretized Gaussian process $\hat{Z}$ that is approximately Markov with zero mean, variance 1, and auto-correlation function $\rho_{\hat{Z}}(k) = E[\hat{Z}_i\hat{Z}_{i+k}] \approx e^{-k/N}$. By well known approximations, it has entropies

$$\begin{aligned} H(\hat{Z}_1) &\approx h(Z(0)) - \log\Delta = \frac{1}{2}\log 2\pi e - \log\Delta \\ H(\hat{Z}_k|\hat{Z}_0) &\approx h\left(Z(\tfrac{k}{N}) \mid Z(0)\right) - \log\Delta = \frac{1}{2}\log 2\pi e(1-e^{-2k/N}) - \log\Delta\,, \end{aligned}$$

where h denotes differential entropy. Accordingly, we assume the X's have the entropies given by the approximate formulas on the right-hand sides of the above.

Figures 3.5 and 3.6 show the rate loss performance of different schemes for $N = 45$ and $N = 1215$, respectively. The encoder loss probability is $p = 0.1$. The error bars show the standard deviation of F. For the clustered SW and Master-Slave schemes, an operating point is plotted for each cluster size K that divides N. As expected, the points in the figure representing No SW and Complete SW are the two extreme points – with the former having highest rate and lowest loss factor, and the latter

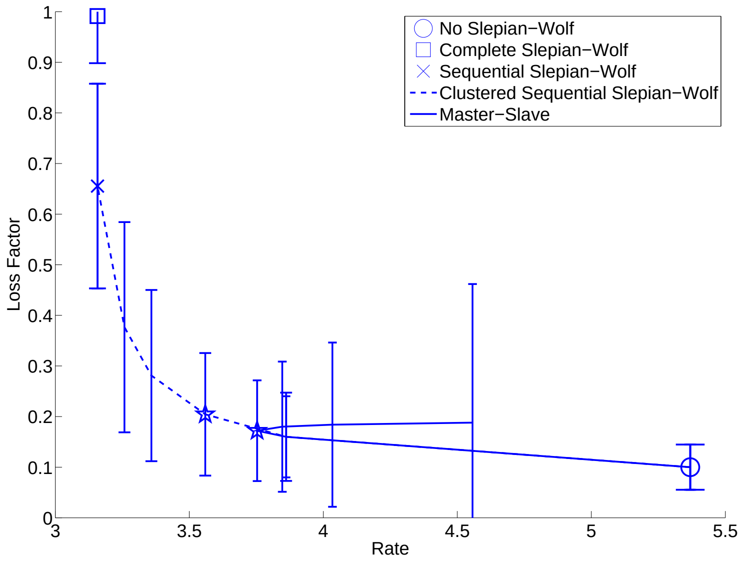


Figure 3.5: Loss factor vs. rate for various schemes for $N = 45$ and $p = 0.1$. Error bars show the standard deviation of the fraction of incorrectly decoded sources. The stars mark intuitively attractive operating points: $K = 5$ for both Clustered Sequential SW and Master-Slave.

having the opposite. For large N, the Sequential SW scheme has essentially the same performance as Complete SW. Clustered Sequential SW has the same performance as the No SW scheme when the cluster size is one and the same performance as Sequential SW when the cluster size is N. Indeed, as the cluster size increases from 1 to N, its rate decreases and its loss factor increases, providing a smooth trade-off between the No SW and Sequential SW.

The Master-Slave scheme follows the Clustered Sequential SW performance from cluster size one up to a certain point, and then its loss factor becomes smaller than that of Clustered Sequential SW for a small range of rates. As the cluster size increases further, the rate of the Master-Slave scheme actually increases. This happens because the rate is the average of the $H(X_1)$ and the smaller, but increasing, terms $H(X_2|X_1), H(X_3|X_1),...H(X_K|X_1)$. Therefore, when K is small, incrementing it adds a new term $H(X_{K+1}|X_1)$ that is less than the average, and consequently, reduces rate. However, as K increases, $H(X_{K+1}|X_1)$ eventually becomes larger than the average of

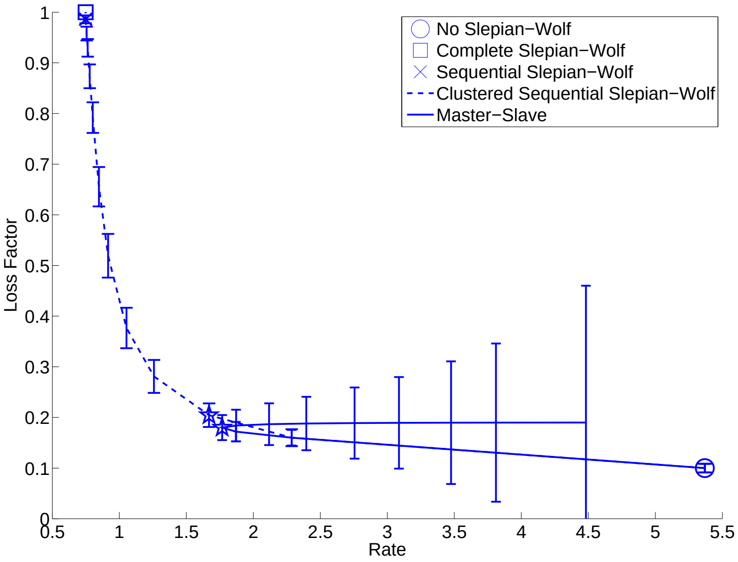


Figure 3.6: Loss factor vs. rate for various rigid schemes for $N = 1215$ and $p = 0.1$. Error bars show the standard deviation of the fraction of incorrectly decoded sources. The stars mark intuitively attractive operating points: $K = 5$ for Clustered Sequential SW, and $K = 9$ for Master-Slave.

the previous terms, so that incrementing K increases rate.

Finally, the error bars in the figures show the standard deviation of F, the fraction of sources that are incorrectly decoded. Comparing the two figures, we see that as the number of sources increases from $N = 45$ to $N = 1215$, the standard deviation decreases for every scheme. Moreover, for all schemes except Master-Slave with large unattractive cluster sizes, when $N = 1215$, the standard deviation of F is fairly small indicating that loss factor $L = E[F]$ by itself is a good measure of reliability, which is what one would expect from the law of large numbers.

3.7 Flexible Coding

To introduce the idea of flexible coding, consider, for example, distributed encoding for three sources at rates $R_1 = H(X_1), R_2 = H(X_2|X_1), R_3 = H(X_3)$. In this case a rigid system can be structured so that X_2 can be decoded if the encodings of both X_1 and X_2 are received. However, since stationarity implies $H(X_2|X_3) = H(X_2|X_1) = R_2$, another rigid system could be structured so that source X_2 can be decoded if the encodings of both X_2 and X_3 are received. However, using Theorem 27 at the above mentioned rates, the encoders and decoders can also be constructed so that source X_2 can be decoded if either the encodings of both X_1 and X_2 are received, or those of both X_2 and X_3 are received. Clearly, such a scheme is more resilient to encoder failures than either rigid scheme.

Now consider flexible coding for N sources $X_1,...,X_N$. On the one hand, as with rigid systems, a flexible coding system has N encoders $e_1,...,e_N$ and N decoders $d_1,...,d_N$, whose specification and operation are the same as with rigid coding. On the other hand, instead of having one dependence set for each source, with flexible coding, each source i has some number, M_i, of dependence sets $\{S_{i,1},...,S_{i,M_i}\}$, and the encoders and decoders are structured so that for each i, source i can be decoded correctly with high probability if the ith decoder receives the encodings of all sources in any one of dependence sets $\{S_{i,1},...,S_{i,M_i}\}$.

Let us call $\mathcal{S}_i = \{S_{i,1},...,S_{i,M_i}\}$ the *dependence set collection* for source i and call $\underline{\mathcal{S}} = \{\mathcal{S}_1,...,\mathcal{S}_N\}$ the *flexible dependence structure*. In order to formalize the idea of decoding from a flexible dependence structure, we say a code *conforms* to a flexible dependence structure $\underline{\mathcal{S}}$ if for each i and each $k \in \{1,...,M_i\}$, there exist functions $\bar{d}_{i,k} : \times_{j \in S_{i,k}} \{1,...\lceil 2^{mR_j} \rceil\} \rightarrow \mathcal{X}^m$ such that

$$d_i(\hat{b}_1,...,\hat{b}_N) = \begin{cases} \bar{d}_{i,k}(\hat{b}_{S_{i,k}}), & \begin{array}{l}\text{where } k \text{ is the smallest integer such that} \\ I_j = 1 \text{ for all } j \in S_{i,k}, \text{ if there is one}\end{array} \\ \Phi_i, & \text{if there is no such } k \end{cases} . \tag{3.8}$$

Note that specifying the smallest k in the above definition is arbitrary; however, some choice is necessary.

We now define a *flexible distributed coding* system s for $X_1,...,X_N$ to be a flexible dependence structure $\underline{\mathcal{S}} = (\mathcal{S}_1,..., \mathcal{S}_N)$ and an $(R_1,..., R_N, m)$ code that conforms to $\underline{\mathcal{S}}$. One can also view a flexible system s with dependence structure $\underline{\mathcal{S}} = \{\mathcal{S}_1,..., \mathcal{S}_N\}$ as a collection of rigid systems $\{s(\sigma) : \sigma \in \mathcal{S}_1 \times ... \times \mathcal{S}_N\}$. Specifically, for each $\sigma = \{S_{1,k_1},..., S_{N,k_N}\}$, $s(\sigma)$ is a rigid system with dependence structure σ and a code that conforms to σ with the same encoders as s and decoders $d_{\sigma,1},..., d_{\sigma,N}$, characterized in the form of (3.1), by the functions

$$\tilde{d}_{\sigma,i}(\hat{b}_{S_{i,k_i}}) = \bar{d}_{i,k_i}(\hat{b}_{S_{i,k_i}})\,.$$

A flexible dependence structure $\underline{\mathcal{S}}$ can be visualized with a *generalized dependence graph*, which is a labeled bipartite graph $G = (V_1, V_2, E)$, where $V_1 = V_2 = \{1,..., N\}$ are just as for the graph representing a rigid dependence structure, and $E = \{(i, j, k) : j \in S_{i,k}\}$ is the set of labeled edges, with (i, j, k) denoting an edge from node i to node j labeled by k, indicating that j is needed in order to decode X_i with the kth dependence set $S_{i,k}$.

3.7.1 Performance and achievability

As with rigid systems, the principal performance measures for a flexible system s are average rate $R_{av}(s)$ and loss factor $L(s, p)$, as defined by (3.2) and (3.3). The loss factor of a system s with flexible dependence structure $\underline{\mathcal{S}}$ decomposes in a somewhat more complicated manner than (3.4):

$$L(s, p) = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{M_i} P_{e,i,k}\,\Pr(A_{i,k}) + \frac{1}{N}\sum_{i=1}^{N}\Pr(A_i^c)\,, \tag{3.9}$$

where $A_{i,k}$ is event that decoding of X_i is done with $\bar{d}_{i,k}$ (see (3.8) for the conditions for such), $P_{e,i,k}$ is the probability of erroneously decoding source i given $A_{i,k}$, and $A_i = \cup_{k=1}^{M_i} A_{i,k}$. The first term on the right side above can be upper bounded as follows:

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{M_i} P_{e,i,k}\Pr(A_{i,k}) \leq \frac{1}{N}\sum_{i=1}^{N}\max_k P_{e,i,k} \triangleq L_c(s,0)\,.$$

Since for a rigid code, $L_c(s,0)$ reduces to $L(s,0)$, we consider $L_c(s,0)$ to be a more general definition of *code loss factor*. The second term on the right side of (3.9) is defined to be the *dependence structure loss factor* $L_d(\underline{\mathcal{S}},p)$. Using all of the above gives a bound analogous to (3.7):

$$L_d(\underline{\mathcal{S}},p) \leq L(s,p) \leq L_d(\underline{\mathcal{S}},p) + L_c(\underline{\mathcal{S}},0)\,,$$

Unlike (3.5), there is no simple general expression for $\Pr(A_i)$ or, consequently, $L_d(\underline{\mathcal{S}},p)$. However, one can see that adding additional dependence sets to $\mathcal{S}_i$ generally increases $\Pr(A_i)$, and consequently, reduces the loss factor.

Again as with rigid systems, we focus on lossless codes and define *achievable rate-vector* $(r_1,...,r_N)$, *achievable rate region* $\mathcal{R}(\underline{\mathcal{S}})$ and $R_{min}(\underline{\mathcal{S}})$ just as in Definition 22, but with $L(s,0)$ replaced by $L_c(s,0)$, and $\underline{S}$ replaced by $\underline{\mathcal{S}} = \{\mathcal{S}_1,...,\mathcal{S}_N\}$. An inner bound to $\mathcal{R}(\underline{\mathcal{S}})$ is the set $\mathcal{R}_{SW,f}(\underline{\mathcal{S}})$ consisting of all rate vectors satisfying the constraints of Theorem 27. Similarly, an upper bound to $R_{min}(\underline{\mathcal{S}})$ can be calculated using $\mathcal{R}_{SW,f}(\underline{S})$ instead of $\mathcal{R}(\underline{S})$. Note that for a rate-vector to be achievable, it must be achievable for each dependence structure in $\mathcal{S}_1 \times ... \times \mathcal{S}_N$. One might think that such multiple achievability constraints will cause $R_{min}(\underline{\mathcal{S}})$ to be larger than for a rigid system. However, in the three-source system introduced at the beginning of this section and in the schemes presented in the next, additional dependence sets are added to a rigid system without increasing $R_{min}(\underline{\mathcal{S}})$, but nevertheless decreasing the loss factor, due to the fact that every additional dependence set gives the decoder an additional chance to correctly decode.

For flexible systems, we define the rate-loss region, $\mathcal{RL}_f$ and the loss-rate function,

$\mathcal{L}_f(r)$, just as in Definitions 23 and 25. Again, $\mathcal{L}_f(r)$ is a north-east quadrant shown in Lemma 24.

3.8 Several Flexible Schemes

We now introduce three families of flexible systems, i.e., schemes.

1) Consecutive-K scheme

In this scheme the rates and encoders are chosen so that for each i successful decoding of source i is possible with high probability if the encodings are received from any set of K consecutive sources that includes i. In other words, the constraints sets for i have the form

$$S_{i,k} = \{i - K + k, ..., i, ..., i - 1 + k\}\,, \quad k = \max\{1, K + 1 - i\}, ..., \min\{K, N + 1 - i\}\,.$$

Assuming the sources form a Markov chain, then just as with Clustered Sequential SW, it can be shown using Theorem 27 that each encoder can have rate

$$R_i = \frac{1}{K} H(X_1, ..., X_K)\,,$$

which is the minimal rate if all encoders are constrained to have the same rate. As shown in Appendix B, the loss factor is

$$L_d(\underline{\mathcal{S}}, p) = 1 - (1-p)^K (1 + (K-1)p)\,.$$

Note that the average rate of this scheme is the same as that of Clustered Sequential SW rigid coding with cluster size K, and the loss factor is smaller for small to moderate values of K, and larger for large values of K. Since all nodes transmit at equal rates, as K increases to N, the loss factor and rate of this scheme approach that of the Complete SW scheme, which is worse than that of Clustered Sequential SW. However, for smaller values of K, the decoder for every node gets multiple shots at decoding, which causes the loss factor of consecutive-K scheme to be less than that of Clustered Sequential SW for small values of K.

2) Clustered-K-choose-n scheme

In this scheme the sources are partitioned into clusters of size K, and the rates and encoders are chosen so that for each i successful decoding of source i is possible with high probability if the encodings are received from the encoding of i as well as any set of $n-1$ other encoders in the same cluster.

Assuming the sources form a Markov chain, then the result in [11, Section III] shows that the joint entropy of n out K stationary sources is maximized when the n sources are uniformly spread over all K sources. Combining this result with Thm. 27, it can be shown that it suffices for each encoder to use rate

$$R_i = \max\left\{\frac{1}{n}\Big(H(X_1) + cH(X_{b+1}|X_1) + (n-c-1)H(X_{a+1}|X_1)\Big)\ ,\ H(X_1|X_{K-n+2})\right\},$$

where $a = \lfloor \frac{K-1}{n-1} \rfloor$, $b = \lceil \frac{K-1}{n-1} \rceil$ and $c = K-(n-1)a-1$. Once again this rate is minimal if all encoders are constrained to have the same rate. As shown in Appendix B, the loss factor is

$$L_d(\underline{\mathcal{S}}, p) = p + \frac{1}{K}\sum_{j=0}^{n-1} j \left(\begin{array}{c} K \\ j \end{array}\right) (1-p)^j p^{K-j}$$

For K fixed, it can be seen that as n increases, the rate R_i decreases and the loss factor increases. Moreover, when K is large, the law of large numbers indicates that with high probability the fraction of encoder failures will be close to p. Hence, if $n < (1-p)K$, with high probability at least n encodings will be received, and the loss factor will be close to the minimal value of p. On the other hand if $n > (1-p)K$, then with high probability the number of received encodings will be less than n and the loss factor will be close to maximal value of 1. Because of this and the fact that rate decreases with n, it makes sense to choose n just a little less than $(1-p)K$.

3) Consecutive-K-choose-n scheme

In this scheme, which combines elements of the previous two, the rates and encoders are chosen so that for each i, successful decoding of source i is possible with high probability if the encodings are received from the encoding of i as well as those of any $n-1$ other sources j, such that $|j-i| \leq K$.

Since decoding is similar to that of Clustered-K-choose-n scheme in the sense that we need bits from at least n sources in a cluster size of K to decode, the encoding rates for this scheme can be the same as for the previous scheme. Since this scheme will successfully decode under more conditions than the previous, it has a lower loss factor. Unfortunately, we have not been able to obtain a useful analytical expression for the loss factor. However, we are able to compute the loss factor by simulation of the system.

The rate and loss factor of this scheme behave in a similar fashion as that of clustered K-choose-n. Thus, for large K, following the arguments in the previous scheme, a good choice of n would be slightly less than $p(1-K)$.

3.9 Comparison of Flexible Schemes

In this section we compare the performances of the rigid and flexible coding schemes on the same sources as in Section 3.6. Figures 3.7 and 3.8 plot loss factor vs. rate for all the rigid and flexible coding schemes for $p = 0.1$ and $N = 45$ and $N = 1215$, respectively. For the Consecutive-K scheme, a point is plotted for each K from 1 to N. For Clustered-K-choose-n the performance for all possible (K, n) pairs was computed, and only those points on the lower convex hull are plotted; likewise

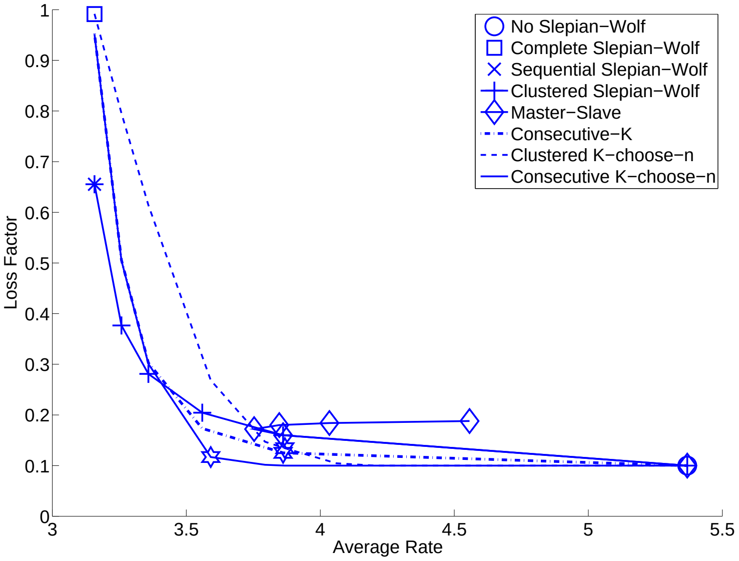


Figure 3.7: Loss factor vs. rate for various schemes for $N = 45$ and $p = 0.1$.

for Consecutive-K-choose-n.

We see that in the important regime of small to moderate loss factor, p to $3p$, flexible schemes clearly outperform their rigid counterparts, having significantly lower average rates. On the other hand, in the less important regime of high loss factor, the rigid Clustered Sequential SW scheme has the best performance, i.e., lowest rate, although the improvement over the best flexible schemes is not terribly large, at most 20%. In addition, the flexible schemes come close to achieving the smallest possible loss factor, p, with average rate significantly less than $H(X_1)$. In fact the Consecutive-K-choose-n scheme attains loss factor approximately p over more than 75% of the achievable rate range of $(\frac{1}{N}H(X_1,...,X_N), H(X_1))$. It should not be surprising that the Consecutive-K-choose-n scheme is the best of the flexible schemes, because (a) it performs at least as well as Consecutive-K, because the latter is a special case of the former, and (b) for any given K and n, it has the same rate but smaller loss factor than Clustered-K-choose-n.

Finally, in an attempt to identify attractive operating points, we identify values

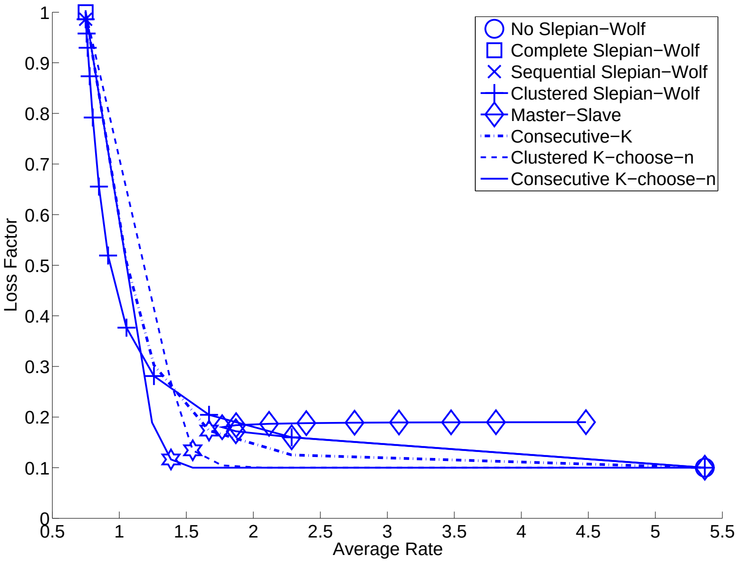


Figure 3.8: Loss factor vs. rate for various schemes for $N = 1215$ and $p = 0.1$.

of K and n at the knees of the loss-rate curves for the flexible schemes (marked by stars on Figures 3.7 and 3.8), where by "knee" we mean the point below which even a small reduction in rate (10-15%) causes the loss factor to increase significantly (40% or more). For both $N = 45$ and 1215 the (K, n) values are (5,n.a.), $(9, 7)$ and $(9, 8)$ for Consecutive-K, Clustered-K-choose-n and Consecutive-K-choose-n schemes, respectively. The corresponding rate-loss values as $(3.55, 0.17)$, $(3.86, 0.13)$ and $(3.59, 0.12)$ for $N = 45$, and $(1.67, 0.17)$, $(1.54, 0.13)$ and $(1.38, 0.12)$ for $N = 1215$. As expected, the optimal choice of n for both Clustered-K-choose-n and Consecutive-K-choose-n is slightly smaller than $(1 - p)K$.

3.10 Concluding Remarks

This chapter introduced the problem of distributed coding in the presence of encoder failures. It provided a theoretical formulation, with loss factor and average rate as the performance measures. It extended the Slepian-Wolf theorem to the case that some encoder outputs are not received at the decoder. It introduced and analyzed several rigid and flexible schemes, both as concrete approaches to trading rate for increased reliability, and as bounds to the achievable rate-loss region. It evaluated the performance of these schemes on a spatial Markov chain of N temporally IID sources, with parameters chosen to model the sampling and uniform quantization of an underlying Gauss-Markov process. It was found that flexible schemes are the only ones that achieve the smallest possible loss factor at rates significantly smaller than if all sources were to be encoded independently.

With a view towards future work, we note that all schemes considered have an underlying structure — rigid or flexible. One can also envision a *fully flexible* scheme comprising a set of encoders and decoders with no underlying dependence structure. On receiving the indices from a set of encoders, the decoder would attempt to decode as many sources as possible. For example, if a binning-based encoder and jointly typical decoder were used, such as described in the proof of Theorem 27, then any subset of sources whose encoding rates satisfy the extended Slepian-Wolf conditions could be decoded with high probability, without any knowledge of the encodings of other sources. It might or might not turn out that such a fully flexible scheme offers substantial improvement over the schemes considered in this chapter.

3.A Proof of Lemma 24

We first claim that $(R_{min}(\underline{S}), L_d(\underline{S}, p))$ is achievable for $\underline{S}$, from which it follows that all rate-loss pairs in the north-east quadrant are achievable. Consider a sequence ϵ_n such that $\lim_{n\to\infty} \epsilon_n = 0$. From the definition of $R_{min}(\underline{S})$, for each n there exists a sequence of systems $s_k^{(n)}$, $k = 1, 2,...,$ each with dependence structure $\underline{S}$, such that $R_{av}(s_k^{(n)}) \overset{k\to\infty}{\leq} R_{min}(\underline{S}) + \epsilon_n$ and $L(s_k^{(n)}, 0) \overset{k\to\infty}{\longrightarrow} 0$. Now for each positive integer n, let s_n be $s_k^{(n)}$ for some k large enough that $R_{av}(s_k^{(n)}) \leq R_{min}(\underline{S}) + 2\epsilon_n$ and $L(s_k^{(n)}, 0) \leq \epsilon_n$. Then for this sequence of systems $\limsup_{n\to\infty} R_{av}(s_n) \leq R_{min}(\underline{S})$, and since $L(s_n, p) \leq L_d(\underline{S}, p) + L(s_n, 0)$, we have $L(s_n, p) \overset{k\to\infty}{\leq} L_d(\underline{S}, p)$. This shows $(R_{min}(\underline{S}), L_d(\underline{S}, p))$ is achievable, as claimed.

We conclude by showing that if (r, l) is achievable, then $r \geq R_{min}(\underline{S})$ and $l \geq L_d(\underline{S}, p)$. If (r, l) is achievable with respect to $\underline{S}$, then there exists a sequence of systems s_k such that $R_{av}(s_k) \overset{k\to\infty}{\leq} r$, $L(s_k, 0) \overset{k\to\infty}{\longrightarrow} 0$, and $L(s_k, p) \overset{k\to\infty}{\leq} l$. Since $L(s_k, 0) \overset{k\to\infty}{\longrightarrow} 0$, then from the definition of $R_{min}(\underline{S})$, $r \geq \limsup_{k\to\infty} R_{av}(s_k) \geq R_{min}(\underline{S})$. Also, $L(s_k, p) \geq L_d(\underline{S}, p)$ for all k. Thus $l \geq \limsup_{k\to\infty} L(s_k, p) \geq L_d(\underline{S}, p)$.

3.B Proof of Thm. 27 (Extended Slepian-Wolf)

Given a rate vector $\underline{r} = (r_1,..., r_N)$, let $\Sigma(\underline{r})$ denote the set of all (i, S) supported by $\underline{r}$, i.e.,

$$\Sigma(\underline{r}) = \{(i, S) : r_U \geq H(X_U|X_{S-U}) \text{ for all } U \subset S \text{ such that } i \in U\},$$

Let δ_n be a non-negative sequence converging to zero.

To prove the theorem, we will show that for any n, there exists M such that for all $m \geq M$, there exists a set of encoders $(e_1,..., e_N)$ having blocklengths m and rates $R_i = r_i + \delta_n$, $i \in \{1,..., N\}$, and a set of decoders $\{d_{i,S} : (i, S) \in \Sigma(\underline{r})\}$ such that

$$\tilde{P}_e \triangleq \sum_{(i,S)\in\Sigma(\underline{r})} P_e(e_S, \tilde{d}_{i,S}) < \delta_n \tag{3.10}$$

where $\tilde{P}_e$ is called the *overall probability of error.* The existence of a sequence of systems satisfying the specifications of the theorem follows directly from this.

To design the encoders and decoders, we will use the conventional random coding approach [3] with binning-type encoding and jointly typical decoding, and show that it has the stronger property (3.10).

For any $S \subset \{1,..., N\}$ and any positive integer m, define the jointly typical set of sequences $x_S^m = \{x_k^m : k \in S\}$ as

$$A_\epsilon^m(S) = \left\{ x_S^m : \left| -\frac{1}{m} \log p(x_T^m) - H(X_T) \right| < \epsilon \text{ for all } T \subset S \right\} ,$$

where $\epsilon \triangleq \delta_n/3$.

Random codebook: For $i = 1, 2,..., N$ and each $x^m \in \mathcal{X}^m$, assign a random index $J_i(x^m) \in \{1, 2,..., \lceil 2^{mR_i} \rceil\}$. The index assignment is done independently for each sequence x^m and uniformly on $\{1, 2,..., \lceil 2^{mR_i} \rceil\}$. The set of x^m for source i assigned to index j is called the jth *bin* for source i.

The random codebook, denoted $\tilde{J}$, is then the collection of all these random indices, i.e., $\tilde{J} = \{J_i(x^m) : i = 1, 2,..., N, \;\; x^m \in \mathcal{X}^m\}$. Let $\tilde{j} = \{j_i(x^m) : i = 1, 2,..., N, \;\; x^m \in \mathcal{X}^m\}$ be an instance of the codebook. We also let $\tilde{j}$ represent a distributed source code with the encoder and decoder described below, and we let $\tilde{P}_e(\tilde{j})$ denote its overall error probability. $\tilde{J}$ denotes a randomly chosen distributed code.

Encoder: Given a codebook $\tilde{j}$, the encoder for the ith source, denoted $e_{\tilde{j},i}$, maps a source sequence $x_i^m \in \mathcal{X}^m$ into the index of the bin to which x_i^m belongs. That is, $e_{\tilde{j},i}(x_i^m) = j_i(x_i^m)$. Let $e_{\tilde{j},S} = \{e_{\tilde{j},i} : i \in S\}$.

Decoder: Given a codebook $\tilde{j}$, the decoder for source i and dependence set S, denoted $d_{\tilde{j},i,S}$, receives a set of indices $l_S = \{l_k : k \in S\}$ and outputs $\hat{x}_i^m$ if it is the only member of $\mathcal{X}^m$ such that the following two conditions hold: (a) $e_{\tilde{j},i}(\hat{x}_i^m) = l_i$ and (b) there exists $x_{S-\{i\}}^m$ such that $e_{\tilde{j},k}(x_k^m) = l_k$ for $k \in S - \{i\}$ and $(\hat{x}_i^m, x_{S-\{i\}}^m) \in A_\epsilon^m(S)$.

Probability of error: In order to show the existence of a code $\tilde{j}$ with overall error probability $\tilde{P}_e(\tilde{j})$ less than δ_n as in (3.10), we will show that for all sufficiently large m,

$$\bar{\tilde{P}}_e < \delta_n \tag{3.11}$$

where

$$\bar{\tilde{P}}_e = \sum_{\tilde{J}=\tilde{j}} \Pr(\tilde{J} = \tilde{j}) \tilde{P}_e(\tilde{j})$$

is the *average* overall error probability of the random codes $\tilde{J}$, averaged over all possible codebooks, i.e all possible $\tilde{j}$. It will then follow immediately that there exists at least one codebook $\tilde{j}$ with $\tilde{P}_e(\tilde{j}) < \delta_n$ and the proof will be complete.

Let $P_e(\tilde{j}, i, S) \triangleq P_e(e_{\tilde{j},S}, d_{\tilde{j},i,S})$ denote the probability of error of the code $\tilde{j}$ for source i and dependence set S. Then,

$$\begin{aligned}\bar{\tilde{P}}_e &= \sum_{\tilde{j}} \Pr(\tilde{J} = \tilde{j}) \sum_{(i,S)\in\Sigma(\underline{r})} P_e(\tilde{j}, i, S) = \sum_{(i,S)\in\Sigma(\underline{r})} \sum_{\tilde{j}} \Pr(\tilde{J} = \tilde{j}) P_e(\tilde{j}, i, S) \\ &= \sum_{(i,S)\in\Sigma(\underline{r})} \bar{P}_e(i, S) \end{aligned} \tag{3.12}$$

where $\bar{P}_e(i, S) \triangleq \Pr(X_i^m \neq d_{\tilde{J},i,S}(e_{\tilde{J},S}(X_i^m))$ is the probability of error in a random experiment in which the sources produce random $X_S^m = \{X_j^m, j \in S\}$, a random code $\tilde{J}$ is chosen, the code is used to encode X_S^m with $e_{\tilde{J},S}$ and to decode source i with $d_{\tilde{J},i,S}$ producing $\hat{X}_i^M$. We emphasize that the randomness comes in two places: the sources in S and the random codebook $\tilde{J}$.

We now come to the heart of the proof, which is to show there exists M such that when $m > M$,

$$\bar{P}_e(i, S) < \frac{\delta_n}{|\Sigma(\underline{r})|} \tag{3.13}$$

for all $(i, S) \in \Sigma(\underline{r})$. Substituting this into (3.12) will demonstrate that $\bar{\tilde{P}}_e < \delta_n$, which will establish (3.11) and complete the proof.

Accordingly, consider some $(i, S) \in \Sigma(\underline{r})$. When an error occurs in decoding source i with dependence set S, i.e., when $X_i^m \neq d_{\tilde{J},i,S}(e_{\tilde{J},S}(X_i^m))$, the error in decoding source i can either be *declared* or *undeclared*. A declared error, denoted E_d, occurs when either (1) there is no x_i^m satisfying both (a) and (b), or (2) there are two or more x_i^m satisfying (a) and (b). We denote the two kinds of declared errors by $E_{d,1}$ and $E_{d,2}$. An undeclared error, denoted E_u, occurs when there is one and only one x_i^m satisfying both (a) and (b), but it does not equal X_i^m, the actual output of the ith source. Note that $E_{d,1}, E_{d,2}, E_u$ depend implicitly on i and S.

One can straightforwardly verify that if $E_{d,1}$ or E_u occurs, then the actual source outputs are not jointly typical, i.e., $X_S^m \notin A_\epsilon^m(S)$. Therefore,

$$\begin{aligned}\bar{P}_e(i,S) &= \Pr(E_{d,1} \cup E_{d,2} \cup E_u) \\ &\leq \Pr((A_\epsilon^m)^c \cup E_{d,2}) \\ &= \Pr((A_\epsilon^m)^c) + \Pr(\underbrace{E_{d,2} \cap A_\epsilon^m}_{\tilde{E}_{d,2}}) \,. \end{aligned} \tag{3.14}$$

Consider. the first term in (3.14). The asymptotic equipartition property (AEP) implies that there exists $M_0 > 0$ such that for all $m > M_0$

$$\Pr((A_\epsilon^m)^c) < \frac{\delta_n}{2|\Sigma(\underline{r})|} \,. \tag{3.15}$$

Indeed, since there are only finitely many choices of (i, S) in $\Sigma(\underline{r})$, we can choose M_o so large that this holds for all $(i,S) \in \Sigma(\underline{r})$.

Now consider the last term in (3.14). Since $\tilde{E}_{d,2} \subset A_\epsilon^m$, the occurrence of $\tilde{E}_{d,2}$ implies that X_1^m satisfies (a) and (b). It follows that

$$\tilde{E}_{d,2} = \{\, \exists\, x_S'^m : x_i'^m \neq X_i^m, e_{\tilde{J},k}(x_k'^m) = e_{\tilde{J},k}(X_k^m), \text{ for all } k \in S \text{ and } x_S'^m \in A_\epsilon^m(S)\} \,.$$

We further subdivide this event depending on those $x_j'^m$ that equal the originally occurring X_j^m. For each $U \subset S$ such that $i \in U$ define the error event

$$\tilde{E}_{d,2}(U) = \big\{\, \exists\, x_S'^m \in A_\epsilon^m(S) \,:\, x_k'^m \neq X_k^m,\ e_{\tilde{J},k}(x_k'^m) = e_{\tilde{J},k}(X_k^m),\ \text{all } k \in U, \\ \text{and } x_l'^m = X_l^m,\ \text{all } l \in S-U \,\big\}\ .$$

Then the events $\tilde{E}_{d,2}(U)$ are disjoint, $\tilde{E}_{d,2} = \cup_{U\subset S:\, i\in U}\, \tilde{E}_{d,2}(U)$, and

$$\begin{aligned}
\Pr(\tilde{E}_{d,2}(U)) &= \sum_{x_S^m} p(x_S^m) \Pr\left(\begin{array}{c} \exists\, x_S'^m \in A_\epsilon^m(S) : x_k'^m \neq X_k^m,\ \text{all } k \in U, \\ e_{\tilde{J},U}(x_U'^m) = e_{\tilde{J},U}(X_U^m),\ x_{S-U}'^m = X_{S-U}^m \end{array} \middle|\ X_S^m = x_S^m \right) \\
&= \sum_{x_S^m} p(x_S^m) \Pr\left(\begin{array}{c} \exists\, x_S'^m \in A_\epsilon^m(S) : x_k'^m \neq x_k^m,\ \text{all } k \in U, \\ e_{\tilde{J},U}(x_U'^m) = e_{\tilde{J},U}(x_U^m),\ x_{S-U}'^m = x_{S-U}^m \end{array} \right) \\
&\leq \sum_{x_S^m} p(x_S^m) \sum_{\substack{x_S'^m \in A_\epsilon^m(S): \\ x_k'^m \neq x_k^m,\, k\in U,\, x_{S-U}'^m = x_{S-U}^m:}} \Pr\left(e_{\tilde{J},U}(x_U'^m) = e_{\tilde{J},U}(x_U^m)\right) \\
&\leq \sum_{x_S^m} p(x_S^m)\, 2^{-mR_U} |A_\epsilon^m(S) \cap \{x_S'^m : x_{S-U}'^m = x_{S-U}^m\}| \\
&\leq \sum_{x_S^m} p(x_S^m)\, 2^{-m(r_U+\delta_n)}\, 2^{mH(X_U|X_{S-U})+2m\epsilon} \\
&= 2^{-m(r_U+\delta_n-H(X_U|X_{S-U})-2\epsilon)} \leq 2^{-m\delta_n/3}
\end{aligned}$$

where the second equality follows from the independence of X_S^m and the code $\tilde{J}$, the first inequality uses the union bound, the second inequality comes from the way random indices are assigned and from ignoring the constraints that $x_k'^m \neq x_k^m$, $k \in U$, the third inequality uses a standard argument, and the last inequality uses the facts that $\epsilon = \delta_n/3$ and that $r_U > H(X_U|X_{S-U})$ since $(i,S) \in \Sigma(\underline{r})$, $U \subset S$ and $i \in U$. We can now choose $M > M_o$ so large that the right-hand side of the above is small that for all $m > M$

$$\Pr(\tilde{E}_{d,2}) = \sum_{U \subset S:\, i \in U} \Pr(\tilde{E}_{d,2}(U)) < \frac{\delta_n}{2|\Sigma(\underline{r})|}\,.$$

Substituting this and (3.15) into (3.14) shows (3.13), which completes the proof.

3.C Loss Factor Calculations

1) Consecutive-K

Consider a source i such that $K \leq i \leq N-K+1$ (to ignore edge effects). Let Y_j be the event that encoder i does not fail and Y_j^k be the event that encoders $j, j+1, ..., k$ do not fail. Then source i can be decoded if and only if the following event occurs

$$Y_{i-K+1}^{i} \cup Y_{i-K+2}^{i+1} \cup Y_{i-K+3}^{i+2} \cup ... \cup Y_{i}^{i+K-1}$$

which equals the following union of disjoint events

$$Y_{i-K+1}^{i} \cup (Y_{i-K+1}^{c} \cap Y_{i-K+2}^{i+1}) \cup (Y_{i-K+2}^{c} \cap Y_{i-K+3}^{i+2}) \cup ... \cup (Y_{i-1}^{c} \cap Y_{i}^{i+K-1})$$

where c denotes set complement. Therefore,

$$\begin{aligned}\Pr(\text{source } i \text{ can be decoded}) &= \Pr(Y_{i-K+1}^{i}) + \sum_{j=1}^{K-1} \Pr(Y_{i-K+j}^{c}) \Pr(Y_{i-K+j+1}^{i+j}) \\ &= (1-p)^K + \sum_{j=1}^{K-1} p(1-p)^K = (1-p)^K(1+(K-1)p)\end{aligned}$$

and the loss factor of the system is

$$L_d(\underline{\mathcal{S}}, p) = 1 - (1-p)^K(1+(K-1)p)\,.$$

2) Clustered K-choose-n

Since the loss factor of all clusters is the same, it suffices to compute the loss factor of just one cluster. Let Y represent the number of encoders in a cluster that are active. Then Y is binomially distributed with probability of success $1-p$.

$$\begin{aligned}
L_d(\underline{\mathcal{S}}, p) &= \mathsf{E}[\text{Fraction of sensor values lost}] \\
&= \frac{1}{K}\sum_{j=0}^{n-1} \mathsf{E}[\text{Number of sensor values lost}|Y=j]\Pr(Y=j) \\
&\qquad + \frac{1}{K}\sum_{j=n}^{K} \mathsf{E}[\text{Number of sensor values lost}|Y=j]\Pr(Y=j) \\
&= \frac{1}{K}\sum_{j=0}^{n-1} K\Pr(Y=j) + \frac{1}{K}\sum_{j=n}^{K}(K-j)\Pr(Y=j) \\
&= \frac{1}{K}\sum_{j=0}^{n-1} j\Pr(Y=j) + \frac{1}{K}\sum_{j=0}^{K}(K-j)\Pr(Y=j) \\
&= \frac{1}{K}\sum_{j=0}^{n-1} j\Pr(Y=j) + 1 - \frac{1}{K}\mathsf{E}[Y] \\
&= p + \frac{1}{K}\sum_{j=0}^{n-1} j\binom{K}{j}(1-p)^j p^{K-j}
\end{aligned}$$

References

[1] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471 – 480, Jul. 1973.

[2] T. Cover, "A proof of the data compression theorem of slepian and wolf for ergodic sources (corresp.)," *IEEE Transactions on Information Theory*, vol. 21, no. 2, pp. 226 – 228, Mar. 1975.

[3] T. M. Cover and J. A. Thomas, *Elements of information theory.* New York, NY, USA: Wiley-Interscience, 1991.

[4] D. Marco and D. L. Neuhoff, "Reliability vs. efficiency in distributed source coding for field-gathering sensor networks," in *IPSN*, 2004, pp. 161–168.

[5] S. Coleri and P. Varaiya, "Fault tolerance and energy efficiency of data aggregation schemes for sensor networks," in *IEEE Vehicular Technology Conference, 2004.*, vol. 4, Sep. 2004, pp. 2925 – 2929.

[6] J. Chen and T. Berger, "Robust coding schemes for distributed sensor networks with unreliable sensors," in *IEEE International Symposium on Information Theory.*, Jun. 2004, p. 116.

[7] S. Pradhan, R. Puri, and K. Ramchandran, "n-channel symmetric multiple descriptions - part i: (n, k) source-channel erasure codes," *IEEE Transactions on Information Theory*, vol. 50, no. 1, pp. 47 – 61, Jan. 2004.

[8] V. Prabhakaran, D. Tse, and K. Ramachandran, "Rate region of the quadratic gaussian ceo problem," in *IEEE International Symposium on Information Theory.*, Jun.-Jul. 2004, p. 119.

[9] N.-M. Cheung and A. Ortega, "Flexible video decoding: A distributed source coding approach," in *IEEE 9th Workshop on Multimedia Signal Processing.*, Oct. 2007, pp. 103 –106.

[10] I. Csiszar and J. Korner, "Towards a general theory of source networks," *IEEE Transactions on Information Theory*, vol. 26, no. 2, pp. 155 – 165, Mar. 1980.

[11] D. Julian and T. Cover, "Concavity in time of conditional entropy for stationary markov sources," in *IEEE International Symposium on Information Theory*, 2002, p. 261.

CHAPTER IV

Sensor Placement and Data Gathering for Field Gathering Wireless Sensor Networks

4.1 Introduction

In this chapter we study two problems associated with data collection in field gathering wireless sensor networks. These networks consist of a fixed number of nodes that are deployed over a given network region. The purpose of the network is to measure some underlying physical phenomenon, e. g. temperature, which we refer to as the field. Every node in the network takes a measurement of the field at its location and communicates this observation to a collector. The collector on receiving observations, from some or all of the nodes, makes an estimate of the field over the entire network region.

The first problem that we study is the sensor placement problem. For this problem, we model the underlying field as a stationary random process with a known autocorrelation function. The sensors take observations of this process at their locations, and communicate those observations to the collector. The collector, on receiving observations from all the sensors, makes an estimate of the process over the entire network region. The metric used to measure the performance of the network is mean squared error (MSE), integrated over the network region. Since the process is correlated, the more sensors that are used, the better is the performance in terms of the MSE. However, in many scenarios we can only place a limited number of sensors to take observations of the field. The question then arises is there an optimal design strategy, that is, is there a placement of sensors that minimizes the MSE. In general, the sensor placement problem would be in two dimensions. However, as a first step, we focus on the sensor placement problem in one dimension.

The one-dimensional sensor placement problem has very strong connections to sampling theory. In essence we are are trying to sample a random process at a fixed number of points so as to provide an optimal reconstruction in terms of the mean squared error. In traditional sampling theory, there is a significant body of literature in the case where the underlying signal being sampled and reconstructed is band-limited. The commonly known Shannon-Nyquist theorem [1, 2] (it is also known as the WKS sampling theorem, named after Whittaker [3], Kotelnikov [4] and Shannon) states that if a signal contains no frequencies higher that f Hz, then it can be completely specified by samples taken $1/2f$ seconds apart. That is, if the signal is sampled uniformly at $2f$ Hz than it can be reconstructed perfectly. The Shannon-Nyquist theorem has extensions: n dimensions [5], reconstructing from function and its derivatives [6, 7], non-uniform points [8], and many others. For more on sampling of band-limited signals see [9, 10, 11, 12]. Another similar problem concerns polynomial interpolation of a curve over an interval in space from a fixed number of points. In this case it can be shown that there exist functions for which when interpolating from uniform points the maximum error approaches infinity as the number of points increases [13]. This is known as Runge's phenomenon. It has been shown that sampling at Chebyshev points minimizes this phenomenon [14, Chapter 6].

For sampling band-limited random processes there is a well known extension of the WKS sampling theorem given by Balakrishnan [15]. However, the literature for sampling and estimation of non-bandlimited random processes is significantly more sparse. In most of the literature, the best linear unbiased estimator (BLUE) is used to estimate the entire field from the samples and the error measure considered is the mean squared error. Sacks and Ylviskar [16, 17], Hajek and Kimeldorf [18] and Wahba [19] model the the random process as sum of known regression functions with unknown coefficients and along with a random error term. They provide asymptotically optimal sampling designs for evaluating regression coefficients. Su and Cambanis [20] consider the problem of estimating the entire process in a given region and provide asymptotically optimal sampling designs for Gaussian processes. For optimal surface interpolation, Micchelli and Wahba [21] give a lower bound to mean squared error in terms of the eigenvalues of the covariance function. However, it is not known whether the lower bound is achievable.

A closely related problem in discrete space is known as the subset selection problem [22, 23]. In this problem a large number of sample points can potentially be observed and the objective is to select the best set of k samples in order to estimate the rest of the sample points or some other correlated random variable. This problem in general is known to be NP-complete. Recently, Das and Kempe [24] have provided polynomial time algorithms providing bounds on the mean squared error in terms of elements of the covariance matrix of the observable variables.

The difficulty of finding optimal design strategies for MSE as the distortion measure arises from the presence of the inverse of covariance matrix of the samples in the estimation coefficients and consequently the MSE. Even for simple correlation functions the inverse covariance matrix behavior is highly non-linear and practically intractable. This has also prompted the use of distortion measures other than mean squared error. Cressie [25] and Ko et al [26] study the optimal design problem for maximizing entropy of the sampled variables. Guestrin et al. [27] examined the optimal sensor placement problem with an objective of maximizing mutual-information.

In both these cases, it is shown that finding the optimal design is an NP-complete problem. However, both entropy and mutual information fall under a class of functions that are called sub-modular [28]. Using the results from Nemhauser et al. [29] it has been shown that choosing sample points greedily one at a time is within $(1-1/e)$ of the optimal set.

In this chapter, we add one result for the class of non-bandlimited random processes. We show that for a Markov process with exponential autocorrelation, the optimal placement of sensors that minimizes the mean squared error is uniform.

The second problem that we study in this chapter is real-time data gathering in sensor networks. Consider a field gathering wireless sensor network where the underlying field is modeled as a stationary spatio-temporal process in a given region. A fixed number of sensors is deployed over the network region. We consider a slotted time system such that each sensor observes the field at its location only at discrete times. The observations at each sensor are then transmitted to the collector, which then estimates the entire field in each time slot. In many of the practical scenarios, all of the sensors will not be able to communicate their data to the collector simultaneously. This could be because sensors would be transmitting over the same wireless channels and thus the collector would only be able to receive data from one sensor in one time-slot. There is also a possibility that some of the sensors are too far from the collector and cannot communicate their data to the collector directly. Instead, they use other sensor nodes as relays. In this case, their data suffers from some delay before getting to the collector. If we place a large number of sensors to observe the field, we get good spatial coverage. However, with sensors competing for time-slots to transmit their data to the collector, much of the data will suffer large delay. On the other hand, a smaller number of sensors means that the data from each sensor gets to the collector more quickly at the expense of better spatial coverage. In this chapter we try to determine the impact of sensor density, and spatial and temporal correlation on the error in estimation.

Related work in data gathering wireless networks concerns analyzing throughput scaling and transport capacity [30, 31], improving network lifetime [32, 33], using distributed coding to compress data [34, 35].

4.2 Sensor Placement in One Dimension: Problem Statement

In this section we give a formal problem statement for the problem of sensor placement over a network region. We consider the one-dimensional case. Let $X(s)$, $-\infty < s < \infty$, denote a zero-mean stationary one-dimensional process and let $I = [0, 1]$ denote the interval representing the network region. Let $R_X(s)$ denote the auto-correlation function for X. We wish to place N sensors, that is find N sample locations, in the network region I. We assume that we make the observations at the end points of the interval I and wish to find N additional sample locations. The set of positions of the sensors/sample points, denoted $S = \{s_0, s_1, s_2,..., s_N, s_{N+1}\}$, is called a *s*ampling design. We assume that the positions in S are ordered in an increasing fashion with $s_0 = 0$ and $s_{N+1} = 1$. Also, let $X(S)$ denote the column vector of observations at the locations in S, and let K_S denote the covariance matrix of these samples.

Let $\widehat{X}(s)$ be the minimum mean squared error (MMSE) linear estimate of $X(s)$ from the samples $X(S)$. Then,

$$\widehat{X}(s) = K_{s,S}^T K_S^{-1} X(S), \tag{4.1}$$

where the $K_{s,S} = \mathsf{E}[X(s)X(S)]$ and T denotes transpose. Then the mean squared error at point s, denoted $E(s, S)$, and the overall mean squared error, denoted $MSE(S)$, are

$$
\begin{aligned}
E(s,S) &= R_X(0) - K_{s,S}^T K_S^{-1} K_{s,S} \\
MSE(I,S) &= \int_I E(s,S)ds
\end{aligned}
\tag{4.2}
$$

The optimal sampling design, denoted S^{opt}, is

$$
S^{opt} = \operatorname*{argmin}_S MSE(I,S) \tag{4.3}
$$

4.2.1 One-Dimensional Gauss-Markov Field

In this subsection, we concentrate on optimal placement of sensors when the underlying process $X(s)$ is Markov. In particular we assume that the autocorrelation function is exponential, i.e., $R_X(s) = \exp(-\sigma|s|)$, where $\sigma > 0$ is the correlation parameter. A higher value of σ implies a lower correlation. A well known example of a stationary Markov process with exponential correlation is the stationary Ornstein-Uhlenbeck process [36].

Since the process is Markov, the error at any given point s depends only on the closest sample point to the left and the closest sample point to the right of s. Thus, $E(s,S)$ and MSE can then be written,

$$
\begin{aligned}
E(s,S) &= E(s,\{s_l,s_r\}) \\
MSE(I,S) &= \sum_{j=1}^{N} MSE(I_j,\{s_{j-1},s_j\}),
\end{aligned}
$$

where s_l, s_r denote the closest sample points to the left and right of s and the interval $I_j = [s_{j-1}, s_j]$.

The following lemma states that given the observations at end points of an interval I, the placement of a sensor that minimizes the MSE over the interval is the mid-point of the interval.

Lemma 30 *Given an interval $I = [s_1, s_2]$ and a stationary Markov process $X(s)$, $s \in I$, with correlation function given by $R_X(s) = \exp(-\sigma|s|)$, $\sigma > 0$. Then, for $s^* = (s_1 + s_2)/2$ and any $s \in I, s \neq s^*$,*

$$MSE(I, \{s_1, s^*, s_2\}) < MSE(I, \{s_1, s, s_2\})$$

The proof of the lemma is given in Appendix 4.A.

The next lemma states that the optimal placement of sensors sensing a Markov field with exponential correlation is uniform.

Lemma 31 *Given a stationary Markov process $X(s)$, $s \in [0, 1]$, and $R_x(s) = \exp(-\sigma|s|)$, where $\sigma > 0$ is the correlation parameter, then the optimal sampling design is uniform, i.e.,*

$$S^{opt} = \left\{0, \frac{1}{N+1}, \frac{2}{N+1}, ..., \frac{N}{N+1}, 1\right\}$$

Proof:

We first observe that $MSE(S)$ is continuous function on S, and S is compact, from the extreme value theorem there exists a global minimum over its domain. We now prove the contra-positive to the statement of the theorem. We show that for any given sampling design S that is not uniform, there exists another sampling design S' such that, $MSE(I, S') < MSE(I, S)$.

Let $S = \{s_0, s_1, s_2, ..., s_N, s_{N+1}\}$ be a given non-uniform sampling design such that $0 = s_0 < s_1 < s_2 < ... s_N < s_{N+1} = 1$. Since the design is non-uniform there exists at least one i such that $s_i \neq (s_{i-1} + s_{i+1})/2$. Choose the smallest such i. Construct a

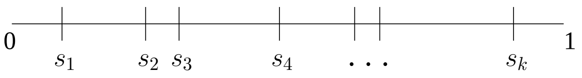


Figure 4.1: Placement of k sensors over the $(0, 1)$ interval.

sampling design $S' = \{s'_j : j = 0, 1, 2..N+1\}$ such that,

$$s'_j = \begin{cases} s_j & j \neq i \\ \dfrac{s_{i-1} + s_{i+1}}{2} & j = i \end{cases}$$

Note that $I_{j-1,j} = I'_{j-1,j}$ for $j \neq i, i+1$ and consequently, $MSE(I_{j-1,j}, \{s_{j-1}, s_j\}) = MSE(I'_{j-1,j}, \{s'_{j-1}, s'_j\})$ for $j \neq i, i+1$. Also, from Lemma 30,

$$MSE(I_{i-1,i}, \{s_{i-1}, s_i\}) + MSE(I_{i,i+1}, \{s_i, s_{i+1}\}) > MSE(I'_{i-1,i}, \{s'_{i-1}, s'_i\}) + MSE(I'_{i,i+1}, \{s'_i, s'_{i+1}\}).$$

Thus,

$$\sum_{j=1}^{N} MSE(I_{j-1,j}, \{s_{j-1,j}, s_{j,j+1}\}) > \sum_{j=1} MSE(I'_{j-1,j}, \{s'_{j-1,j}, s'_{j,j+1}\})$$

$$MSE(I, S) > MSE(I, S')$$

Thus, for any non-uniform sampling design there exists another sampling design that has lower MSE. Combining with the fact that $MSE(S)$ does attain a minimum proves that the uniform sampling design is optimal. □

4.3 Data Gathering in Sensor Networks

In this section, we propose a simple model for data gathering in wireless sensor networks. This model is admittedly a first step in analyzing how spatial and temporal correlation would affect data gathering. We model the underlying field as a two-

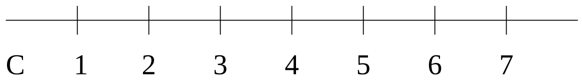


Figure 4.2: Uniform placement of $N = 7$ sensors over the $(0, 1)$ interval. The collector, C, is placed at 0, to the extreme left of all the sensors.

dimensional stationary random process with a known autocorrelation function. One of the dimensions represents space, and the other represents time. We wish to answer the following question: does increasing the number of sensor nodes employed in gathering data from a field of limited size always decrease the MSE, or is there an optimal density of sensors that minimizes the MSE. Initially, one might presume that the larger the number of sensors, the better the reconstruction of the field at the collector. However, as mentioned earlier, due to communication constraints, networks do not allow each sensor to send its data to the collector instantaneously. In fact, in wireless networks using relays, the data from nodes far away from the collector suffers a large amount of delay. The nodes closer to the collector also would suffer some delay. This in general would cause the data to be not as useful in estimating the current value of the field.

We consider a one-dimensional network with N sensors placed uniformly on the real-line between $[0, 1]$. We label these sensors $1, 2, ..., N$ from left to right. Thus the placement of the ith sensor is $s_i = i/N$. The collector is assumed to be placed to the left of the left-most sensor as shown in Figure 4.2. We model the underlying field that the sensors in the network sense by a zero mean stationary random process $X(s, t)$ where s and t denote the spatial and temporal dimensions respectively. The correlation between two samples $X(s_1, t_1)$ and $X(s_2, t_2)$ of the field is given by the autocorrelation function $R(s_1, t_1; s_2, t_2) = R_X(|s_1 - s_2|, |t_1 - t_2|)$.

We assume a slotted time system with each time slot of Δ seconds. Each sensor can both transmit and receive in the same slot. Each sensor communicates its data to the collector by relaying it to the sensor that is immediately to its left. Let $X(s_i, k\Delta)$ denote the the observation of the process by sensor i in time-slot k. Also, let $Y(i, k)$

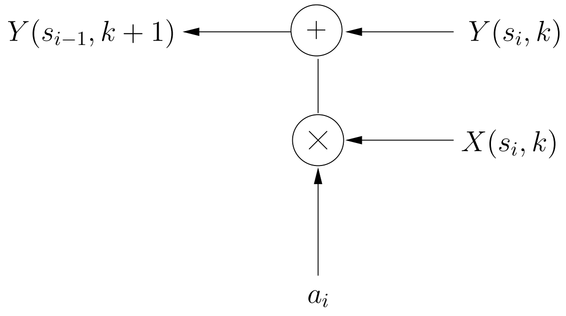


Figure 4.3: The computation at sensor node i. The data sensed in time slot k, $X(s_i, k)$ is scaled by a_i and added to the data received from sensor $i+1$. The sum $Y(i-1, k)$ is transmitted to sensor $i-1$.

denote the data received by sensor i from sensor $i+1$ in time slot k. Then in time-slot $k+1$ the i th sensor transmits $a_i X(s_i, k) + Y(i, k)$ to sensor $i-1$, see Figure 4.3. Thus, in steady state the collector during time slot k receives

$$Y(k) = \sum_{i=1}^{N} a_i X(s_i, k-i) = \boldsymbol{a}^T \boldsymbol{X}_k, \tag{4.4}$$

where $\boldsymbol{a} = [a_1,\ a_2,\ ...\ a_N]^T$ and $\boldsymbol{X}_k = [X(s_1, (k-1)\Delta),\ X(s_2, (k-2)\Delta),\ ...X(s_N, (k-N)\Delta)]^T$ and superscript T indicates transpose. We believe that with each sensor having limited computation capabilities, linear combining of data is a reasonable assumption on combining data. Also, this model could be further extended to the case where for each sensor instead of using the same a_i in each time slot, it can vary a_i's according to a periodic schedule.

The collector at time slot k makes a linear MMSE estimate of the entire field using $Y(k)$ only. Let $\widehat{X}(s, k\Delta) = b_s Y(k)$ be the linear estimate of the field at point s and during time-slot k. Then the mean squared error (MSE) in estimating the entire field is,

$$MSE(N) = \int_0^1 \mathsf{E}[(X(s,k) - \widehat{X}(s,k))^2] ds \tag{4.5}$$

Since for a given N, the sensor placement is fixed and since $\boldsymbol{a}$ does not change with time, the MSE remains same for different time slots.

In order to minimize MSE, it can be easily shown that,

$$\begin{aligned} b_s = \frac{\mathsf{E}[X(s,k\Delta)Y(k)]}{\mathsf{E}[Y(k)^2]} &= \frac{\mathsf{E}\left[X(s,k\Delta)\sum_{i=1}^{N} a_i X(s_i,(k-i)\Delta)\right]}{\mathsf{E}\left[\sum_{i=1}^{N} a_i X(s_i,(k-i)\Delta))^2\right]} \\ &= \frac{\sum_{i=1}^{N} a_i R_X(|s-s_i|, i\Delta)}{\sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j R_X(|s_j-s_i|,|j-i|\Delta)} \\ &= \frac{\sum_{i=1}^{N} a_i R_X(|s-s_i|, i\Delta)}{\boldsymbol{a}^T\Psi\boldsymbol{a}}, \end{aligned} \tag{4.6}$$

where $\Psi(i,j) = R_X(|s_i - s_j|, |i-j|\Delta)$. Note the Ψ is the covariance matrix corresponding to $\boldsymbol{X}_k$.

Then the MSE is,

$$\begin{aligned} MSE(N) &= \int_0^1 \mathsf{E}[(X(s,k\Delta) - \widehat{X}(s,k\Delta))^2]ds \\ &= R_X(0,0) - \int_0^1 b_{s,k}^2 E[Y(k)^2]ds \\ &= R_X(0,0) - \int_0^1 \frac{(\sum_{i=1}^{N} a_i R_X(|s-s_i|, i\Delta))^2}{\boldsymbol{a}^T\Psi\boldsymbol{a}} \qquad (4.7) \\ &= R_X(0,0) - \frac{\sum_{1=1}^{N}\sum_{j=1}^{N} a_i a_j \int_0^1 R_X(|s-s_i|, i\Delta)R_X(|s-s_j|, j\Delta)}{\boldsymbol{a}^T\Psi\boldsymbol{a}} \\ &= R_X(0,0) - \frac{\boldsymbol{a}^T\Phi\boldsymbol{a}}{\boldsymbol{a}^T\Psi\boldsymbol{a}}, \qquad (4.8) \end{aligned}$$

where $\Phi(i,j) = \int_0^1 R_X(|s-s_i|, i\Delta)R_X(|s-s_j|, j\Delta)ds$. Thus, in order to minimize the MSE, we need to maximize $h(\boldsymbol{a}) \triangleq \dfrac{\boldsymbol{a}^T\Phi\boldsymbol{a}}{\boldsymbol{a}^T\Psi\boldsymbol{a}}$. We will show that $h(\boldsymbol{a})$ does indeed have a maximum. For any given, $h(\boldsymbol{a})$ has the same value for any scalar multiple of $\boldsymbol{a}$. Thus, the $\boldsymbol{a}$ obtaining the maximum is not unique. However, our primary concern is the maximum value $h(\boldsymbol{a})$ and a value of $\boldsymbol{a}$ that achieves that maximum.

4.3.1 Optimizing using Lagrange Multipliers

As was stated in the previous section, $h(\boldsymbol{a})$ attains the same value for any scalar multiple of $\boldsymbol{a}$. Thus, in order to find the $\boldsymbol{a}$ that maximizes $h(\boldsymbol{a})$, and the corresponding maximum value of h, we consider the following optimization problem,

$$\max_{\boldsymbol{a}:\boldsymbol{a}^T\Psi\boldsymbol{a}=1} \boldsymbol{a}^T\Phi\boldsymbol{a} \triangleq \max_{g(\boldsymbol{a})=0} f(\boldsymbol{a}), \tag{4.9}$$

where $f(\boldsymbol{a}) = \boldsymbol{a}^T\Psi\boldsymbol{a}$ and $g(\boldsymbol{a}) = \boldsymbol{a}^T\Phi\boldsymbol{a} - 1$. Note that f and h attain their maximum value for the same $\boldsymbol{a}$ and the maximum value of h is the same as that of f constrained to $g = 0$. In order to find the maxima of f, we use the Lagrange multiplier approach. The Lagrangian corresponding to f is,

$$L(\boldsymbol{a}, \lambda) = \boldsymbol{a}^T\Phi\boldsymbol{a} - \lambda(\boldsymbol{a}^T\Psi\boldsymbol{a} - 1) \tag{4.10}$$

In order to find stationary points of L we will require the concept of generalized eigenpairs [37].

Definition 32 *A scalar λ is called* generalized eigenvalue, *for a pair of matrices (A, B) if and only if it satisfies the equation*

$$A\boldsymbol{v} = \lambda B\boldsymbol{v}.$$

The vector $\boldsymbol{v}$ is a generalized eigenvector corresponding to λ.

It can be shown that if A and B are both $n \times n$ Hermitian matrices, then there exist n real generalized eigenvalues. In addition if B is positive definite, then it is possible to find n mutually B-orthonormal eigenvectors $\boldsymbol{v}_i$'s. For more on the generalized eigenvalue problem, see [37, Chapter 4]. Let $((\lambda_1, \boldsymbol{v}_1), (\lambda_2, \boldsymbol{v}_2), ..., (\lambda_N, \boldsymbol{v}_N))$ be a set of generalized eigenpairs for the pair of matrices (Φ, Ψ) such that $\lambda_1 \geq \lambda_2 .. \geq \lambda_N$ and

$\boldsymbol{v}_i$'s are Ψ orthonormal, i.e.,

$$\boldsymbol{v}_i \Psi \boldsymbol{v}_j = \begin{cases} 0 & i \neq j \\ \\ 1 & i = j \end{cases}.$$

Lemma 33 *Given $f(\boldsymbol{x}) = \boldsymbol{x}^T \Phi \boldsymbol{x}$ and $g(\boldsymbol{x}) = \boldsymbol{x}^T \Psi \boldsymbol{x} - 1$, where Φ, Ψ real symmetric and Ψ positive definite, the maximum of $f(\boldsymbol{x})$ constrained to $g(\boldsymbol{x}) = 0$ is the maximum generalized eigenvalue of (Φ, Ψ) and the maximum is attained for any eigenvector corresponding to the maximum generalized eigenvalue.*

The proof of the lemma is given in Appendix 4.B. Thus, using the above lemma the minimum mean squared error for an optimal $\boldsymbol{a}$ is $R_X(0,0) - \lambda_1$.

4.4 Simulations

In this section, we provide some simulation results for the data gathering problem when the underlying process $X(s,t)$ is Gaussian with a separable autocorrelation model that is exponential in both spatial and temporal domains. That is, $R_X(s,t) = \exp(-\sigma|s|)\exp(-\tau|t|)$ for all s, t, where σ and τ are parameters that control spatial and temporal correlations. Note that a higher σ (τ) means a lower spatial (temporal) correlation. We also fix $\Delta = 1$.

For a fixed N, under the separable exponential correlation model, $\Psi(i,j) = R_x(|s_i - s_j|, |i-j|\Delta) = \exp(-\sigma|i-j|/N)\exp(-\tau|i-j|)$. Also, for $\sigma \neq 0$,

$$\begin{aligned}\Phi(i,j) &= \exp(\tau(i+j)) \int_0^1 \exp(-\sigma|s-s_i|)\exp(-\sigma|s-s_j|)ds \\ &= \exp(\tau(i+j)) \left[\frac{1}{\sigma} \exp\left(\frac{|i-j|}{N} \right) \right. \\ &\qquad \left. - \frac{(1+\exp(-2\sigma))}{2\sigma} \exp\left(-\sigma \frac{i+j}{N}\right) + \frac{|i-j|}{N} \exp\left(-\sigma\frac{|i-j|}{N}\right) \right]\end{aligned}$$

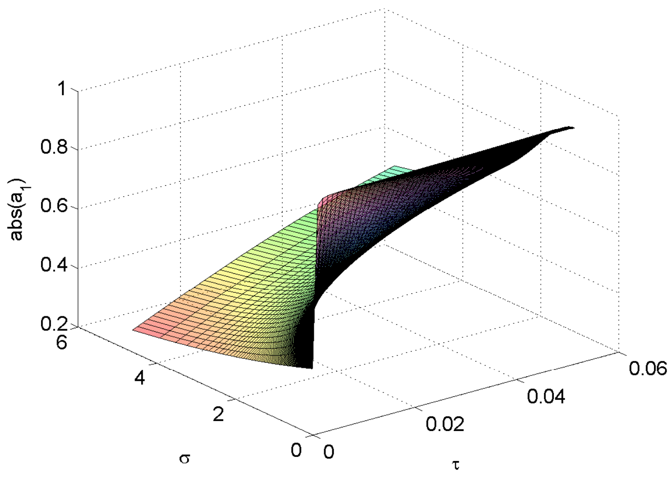


Figure 4.4: Plot of $abs(a_1)$ vs. σ and τ.

For the first part of the simulations we fix the number of sensors to $N = 10$. Figure 4.4 shows a plot of the coefficient of sensor one vs. σ and τ. The weight given to observations from sensor one increases with decreasing temporal correlation or increasing spatial correlation. This is to be expected as when the temporal correlation increases, data from sensors relaying through sensor one even though delayed, is more useful in estimating the current field. Thus, the optimal set of coefficients give a higher weight to data from other sensors. Similarly, as spatial correlation increases, the data from sensor one can give a good estimate of the entire field. Thus, higher the spatial correlation the larger the weight assigned to sensor one.

Figure 4.5 shows a plot of absolute values of the optimal coefficients a_i's for different values of parameter τ. The number of sensors is fixed at $N = 10$ and the spatial correlation parameter is $\sigma = 0.01$. When data is perfectly correlated in time, that is $\tau = 0$, the delay incurred by observations from higher numbered sensors to get to the collector has no effect on estimation of the current value of the field. Thus, the sensors that are highly correlated to the entire line segment $[0, 1]$, that is sensors in the middle, get higher weight. However, as temporal correlation decreases, the delay has more of an effect and the lower the temporal correlation, less useful is the data

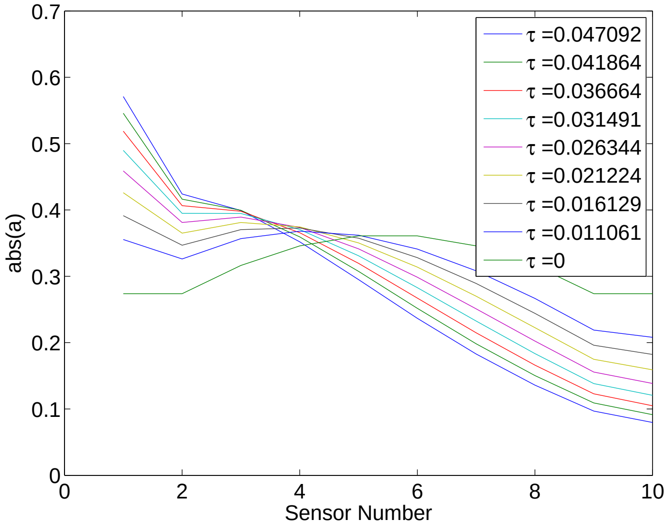


Figure 4.5: Plot of absolute values of a_i's for sensors for different values of τ. $N = 10$ and $\sigma = 0.01$.

from far away sensors. Thus, at low temporal correlation values, the sensors closer to the collector get a higher weight.

Figure 4.6 shows a plot of the absolute values of the optimal coefficients, a_i's, for different values of the spatial correlation parameter σ. The number of sensors is $N = 10$ and the temporal correlation parameter is $\tau = 0.98$. For high spatial correlation, that is small σ, the a_i's decrease with i. This is because at high spatial correlation, the data from sensor one gives a good estimate for the entire field and it suffers minimal delay to get to the collector. At perfect spatial correlation $\sigma = 0$, we would only want to receive data from the first sensor as it incurs the least delay and the entire field can be estimated from it. However, at low spatial correlation, data from other sensors becomes more useful as it gives a better estimate of the field. Thus, for lower values of spatial correlation, data from sensors farther away from the collector is also given higher weights.

Finally, Figure 4.7 shows a plot of the optimal value of N vs. the spatial and temporal correlation parameters. In this figure, on both the X and Y-axes, one represents perfect correlation. Here, for every value of σ and τ, we optimize the

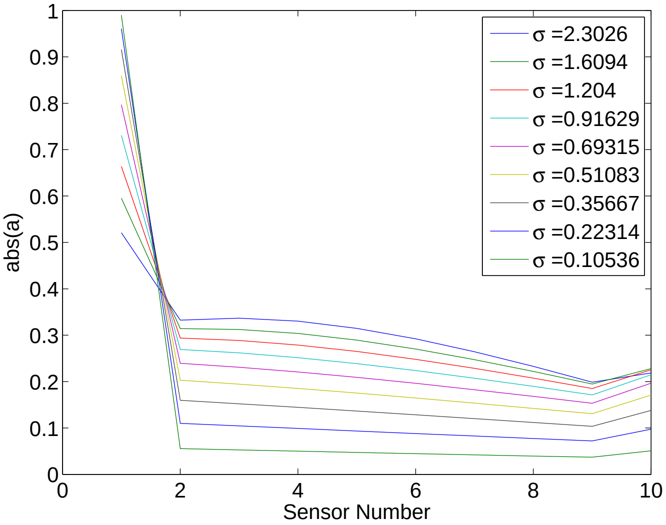


Figure 4.6: Plot of absolute values of a_i's for sensors for different values of σ. $N = 10$ and $\tau = 0.02$.

a_i's for various values of N and then find the N that gives the minimum MSE. As can be seen from the figure, the optimal number of sensors increases if the temporal correlation increases or the spatial correlation decreases. For a fixed value of spatial correlation, as the temporal correlation increases, data that takes multiple hops is still highly correlated to the current value of the field. Thus, a higher number of sensors in this case gives a better estimate of the field. Similarly, for a fixed value of temporal correlation, as the spatial correlation increases, even a small number of sensors can provide a good estimate of the entire field. Thus, the optimal number of sensors decreases with an increase in spatial correlation.

4.A Proof of Lemma 30

Without loss of generality we assume that $I = [0, 1]$. Then $s^* = 0.5$. Also, note that $MSE(s) = MSE(1 - s)$. Thus, in order to show that $MSE(s^*) < MSE(s)$ we need to show that for any $0 < \delta \leq 0.5$

$$MSE(I, \{0, s^*, 1\}) < MSE(I_1, \{0, s^* + \delta, 1\}).$$

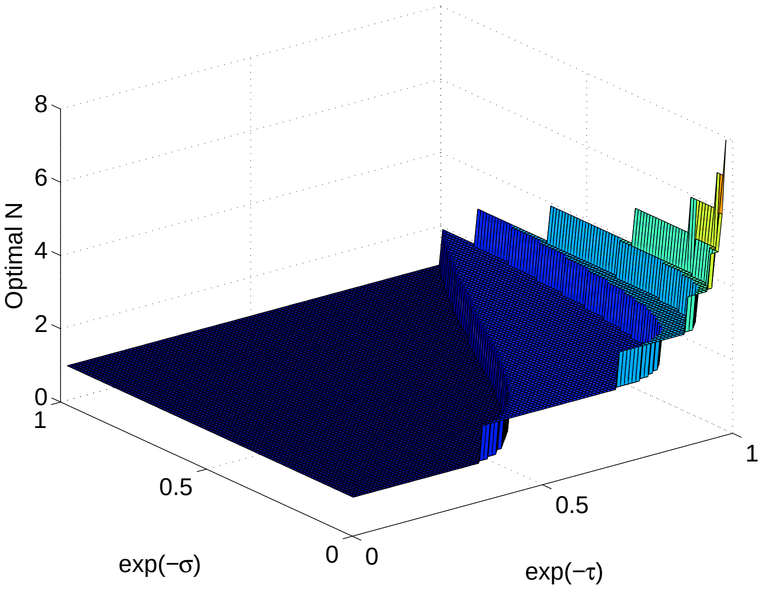


Figure 4.7: Plot of optimal N vs σ and τ.

Using the Markov property, it suffice to show,

$$\int_0^{s^*} E(s,\{0,s^*\})ds + \int_{s^*}^1 E(s,\{s^*,1\})ds < \int_0^{s^*+\delta} E(s,\{0,s^*+\delta\})ds + \int_{s^*+\delta}^1 E(s,\{s^*+\delta,1\})ds. \qquad (4.11)$$

Let $f(t) = \int_0^t E(s,\{0,t\})ds$. Note that $\int_t^1 E(s,\{t,1\})ds = \int_0^{1-t} E(s,\{0,1-t\})dt = f(1-t)$. Then (4.11) can be rewritten as,

$$f(s^*) + f(s^*) < f(s^*+\delta) + f(s^*-\delta).$$

The above equation is a known property of convex functions. In order to show that f is a convex function, we will show that the second derivative of f is strictly positive.

The first derivative of $f(t)$ is

$$f'(t) = \frac{d}{dt}\left(\int_0^t E(s,\{0,t\}\right) ds$$

$$
\begin{aligned}
&= E(t, \{0, t\}) + \int_0^t E'(s, \{0, t\})ds \\
&= \int_0^t E'(s, \{0, t\})ds,
\end{aligned}
$$

where $E'(s, \{0, t\})$ denotes the derivative of $E(s, \{0, t\})$ with respect to t. The second derivative of $f(t)$ is,

$$
\begin{aligned}
f''(t) &= \frac{d}{dt} f'(t) \\
&= E'(s, \{0, t\})|_{s=t} + \int_0^t E''(s, \{0, t\})ds \qquad (4.12)
\end{aligned}
$$

Thus, in order to compute the second derivative of $f(t)$, we need the the first and second derivatives of E.

Recall that $R_X(s) = \exp(-\sigma s)$. Then for $0 \leq s \leq t$,

$$
\begin{aligned}
E(s, \{0, t\}) &= 1 - \frac{1}{1 - e^{-2\sigma t}} (e^{-2\sigma s} - 2e^{-\sigma s} e^{-2\sigma t} e^{-\sigma(t-s)} + e^{-2\sigma(t-s)} \\
E'(s, \{0, t\}) &= \frac{2\sigma e^{-2\sigma t}}{(1 - e^{-2\sigma t})^2} (e^{2\sigma s} + e^{-2\sigma s} - 2) \\
E''(s, \{0, t\}) &= \frac{-4\sigma^2 e^{-2\sigma t}(1 + e^{-2\sigma t})}{(1 - e^{-2\sigma t})^3} (e^{2\sigma s} + e^{-2\sigma s} - 2),
\end{aligned}
$$

where the first equation can be easily obtained using (4.2). Next, we separately evaluate the first and second terms of (4.12)

$$
\begin{aligned}
E(s, \{0, t\})|_{s=t} &= 2\sigma \\
\int_0^t E''(s, \{0, t\})ds &= \frac{-4\sigma^2 e^{-2\sigma t}(1 + e^{-2\sigma t})}{(1 - e^{-2\sigma t})^3} \left(\frac{e^{2\sigma t} - e^{-2\sigma t}}{2\sigma} - 2t \right)
\end{aligned}
$$

Substituting back in (4.12),

$$
f''(t) = 2\sigma - \frac{-4\sigma^2 e^{-2\sigma t}(1 + e^{-2\sigma t})}{(1 - e^{-2\sigma t})^3} \left(\frac{e^{2\sigma t} - e^{-2\sigma t}}{2\sigma} - 2t \right)
$$

$$= \frac{8\sigma e^{-2\sigma t}}{(1-e^{-2\sigma t})^2} \underbrace{\left(-1 + e^{-2\sigma t} + \sigma t(1 + e^{-2\sigma t})\right)}_{g(t)}$$

Thus, if $g(t) > 0$, then $f''(t) > 0$. Using rudimentary differentiation it can be easily shown that $g''(t) > 0$ and $g'(0) = 0$. Thus, $g'(0) > 0$ for all $t > 0$. Combined with the fact that $g(0) = 0$, gives $g(t) > 0$ for all $t \geq 0$. Thus, second derivative of $f(t)$ is positive. This completes the proof of the lemma. □

4.B Proof of Lemma 33

Before going to the proof of Lemma 33 we give a few preliminaries for optimizing using Lagrange multipliers. A detailed discussion of Lagrange optimization and proofs of the lemma's given in this section can be found in [38, Chapter 18].

Let $\boldsymbol{x} = \{x_1,..., x_n\} \in \mathcal{R}^n$, and let $f(\boldsymbol{x}), g(\boldsymbol{x})$ be functions of $\boldsymbol{x}$. Then the gradient and Hessian of $f(\boldsymbol{x})$, denoted $\nabla_{\boldsymbol{x}} f(\boldsymbol{x})$ and $\nabla^2_{\boldsymbol{x}} f(\boldsymbol{x})$, respectively, are defined as follows.

$$\nabla_{\boldsymbol{x}} f(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \ldots & l\frac{\partial f}{\partial x_n} \end{bmatrix}^T.$$

$$\nabla^2_{\boldsymbol{x}} f(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & & & \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix}.$$

For the constrained optimization problem we wish to find $\boldsymbol{x}^*$ such that

$$f(\boldsymbol{x}^*) = \max_{\boldsymbol{x}:g(\boldsymbol{x})=0} f(\boldsymbol{x})$$

We make use of Lagrange multipliers to perform this optimization. The Lagrangian of f is

$$L(\boldsymbol{x}, \lambda) = f(\boldsymbol{x}) - \lambda g(\boldsymbol{x})$$

Let $\nabla L(\boldsymbol{x}, \lambda) = \begin{bmatrix} (\nabla_{\boldsymbol{x}} L)^T & \frac{\partial L}{\partial \lambda} \end{bmatrix}^T$ denote the gradient with respect to $\boldsymbol{x}$ and λ. Also, define the bordered Hessian of L with respect to $\boldsymbol{x}$ as $H_{\boldsymbol{x}}(\boldsymbol{x}, \lambda) = \nabla^2_{\boldsymbol{x}} L(\boldsymbol{x}, \lambda)$ and the set $P(\boldsymbol{x}) = \{\boldsymbol{y} : (\nabla_{\boldsymbol{x}} g)^T \boldsymbol{y} = 0\}$.

The next lemma gives a necessary condition for $\boldsymbol{x}^*$ to be a stationary point of f under the constraint $g = 0$.

Lemma 34 *If $\boldsymbol{x}^*$ is a stationary point of f subject to $g(\boldsymbol{x}) = 0$, then there exists a λ such that $\nabla L(\boldsymbol{x}^*, \lambda) = \underline{0}$.*

Thus, for any stationary point $\boldsymbol{x}$ of f subject to $g = 0$, there exists a λ such that $(\boldsymbol{x}, \lambda)$ is a stationary point of $L(\boldsymbol{x}, \lambda)$.

The next lemma gives a necessary condition for $\boldsymbol{x}^*$ to be a local maximum of f under the constraint $g(\boldsymbol{x}) = 0$.

Lemma 35 (Necessary Conditions) *Let $\boldsymbol{x}^*$ be a local maximum of f subject to $g(\boldsymbol{x}) = 0$. Then there exists a λ such that $\nabla L(\boldsymbol{x}^*, \lambda) = \underline{0}$ and $H_{\boldsymbol{x}}(\boldsymbol{x}^*, \lambda)$ is negative semi-definite on $P(\boldsymbol{x}^*)$.*

We now return to our optimization problem and provide a proof of Lemma 33

Recall that $f(\boldsymbol{x}) = \boldsymbol{x}^T \Phi \boldsymbol{x}$ and $g(\boldsymbol{x}) = \boldsymbol{x}^T \Psi \boldsymbol{x}$, where Φ, Ψ symmetric and Ψ positive definite. Since Ψ is positive definite, the set of points satisfying $g(\boldsymbol{x}) = 0$ is a compact set. Since $f(\boldsymbol{x})$ is a continuous function, from the extreme value theorem, it achieves a global maxima on the set of points satisfying $g(\boldsymbol{x}) = 0$. Thus, if there exist only one local maxima, or multiplicity of local maxima all having the same value, then that value is also the global maxima.

In order the find a point $\boldsymbol{x}$ that achieves the global maximum we show the following. From Lemma 34, for any stationary point $\boldsymbol{x}$ of f subject to $g = 0$ there exist a λ such that $(\boldsymbol{x}, \lambda)$ is a stationary point of L. Thus, we first find all points $(\boldsymbol{x}, \lambda)$ that are stationary points of $L(\boldsymbol{x}, \lambda)$. We then show that any stationary point of $L(\boldsymbol{x}, \lambda)$ which satisfies the necessary condition to be a local maxima, gives the same value for f. Since all local maxima have the same value, and since a global maxima exists, the value of the global maxima is the same as that of any of the local maxima.
Recall that the Lagrangian is

$$L(\boldsymbol{x}, \lambda) = f(\boldsymbol{x}) - \lambda g(\boldsymbol{x}) = \boldsymbol{x}^T \Phi \boldsymbol{x} - \lambda \boldsymbol{x}^T \Psi \boldsymbol{x}.$$

The stationary points of L, given by $\nabla L(\boldsymbol{x}, \lambda) = \underline{0}$ are solutions to the equations

$$\begin{aligned} \Phi \boldsymbol{x} &= \lambda \Psi \boldsymbol{x} \\ \boldsymbol{x}^T \Psi \boldsymbol{x} &= 1. \end{aligned}$$

The solution to above equations are pairs $(\boldsymbol{v}_i, \lambda_i)$ for $i = 1, 2, ..., n$, where λ_i are the generalized eigenvalues of Φ, Ψ and $\boldsymbol{v}_i$'s are corresponding eigenvectors such that they are Ψ orthonormal. Note that, since Ψ is invertible, λ_i's are also the eigenvalues $\Psi^{-1}\Phi$. We order these eigenvalues such that $\lambda_1 \geq \lambda_2 ... \geq \lambda_n$.
In order to verify the necessary conditions of being a local maximum for $(\boldsymbol{v}_i, \lambda_i)$ we will need the following lemma, see [37, Chapter 4], about generalized eigenvectors.

Lemma 36 *Given A, B symmetric real and B positive definite, then there exists a* *set of generalized eigenpairs* (λ_i, v_i)*,* $i = 1,...,n$ *satisfying* $A\boldsymbol{v}_i = \lambda_i B\boldsymbol{v}_i$ *such that*

1. $\boldsymbol{v}_i^T B\boldsymbol{v}_j = 0$ *if* $i \neq j$*.*

2. $\boldsymbol{v}_i^T B\boldsymbol{v}_i = 1$ *for all* i*.*

3. $\boldsymbol{v}_i$*'s for a basis for* $\mathcal{R}^n$*.*

We next show that among the stationary points of L, only $\boldsymbol{v}_1$, an eigenvector corresponding to the largest eigenvalue λ_1, satisfies the necessary conditions. Note that for the given Lagrangian L, the bordered Hessian and $P(\boldsymbol{x})$ are $2\Phi - 2\lambda\Psi$ and $\{\boldsymbol{y} \, : \, \boldsymbol{x}^T\Psi\boldsymbol{y} = 0\}$, respectively. Then for $\boldsymbol{v}_1$, $H_1 \triangleq H(\boldsymbol{v}_1, \lambda_1) = 2\Phi - 2\lambda_1\Psi$ and $P(\boldsymbol{v}) = \{\boldsymbol{y} \, : \, \boldsymbol{v}_1^T\Psi\boldsymbol{y} = 0\}$. We want to show that for all $\boldsymbol{y} \in P(\boldsymbol{v}_1)$, $\boldsymbol{y}^T H_1 \boldsymbol{y} < 0$. Since $\boldsymbol{v}_i$'s form a basis for $\mathcal{R}^n$, then for any $\boldsymbol{y}$ there exist α_i's such that $\boldsymbol{y} = \sum_{i=1}^n \alpha_i \boldsymbol{v}_i$. For $\boldsymbol{y} \in P(\boldsymbol{v}_i)$, using Lemma 36,

$$0 = \boldsymbol{v}_1^T\Psi\boldsymbol{y} = \boldsymbol{v}_1^T\Psi\sum_{i=1}^n \alpha_i\boldsymbol{v}_i = \alpha_i$$

Thus, $\boldsymbol{y} = \sum_{i=2}^n \alpha_i\boldsymbol{v}_i$ and for such a $\boldsymbol{y}$

$$\begin{aligned}
\boldsymbol{y}^T H_1 \boldsymbol{y} &= 2\boldsymbol{y}^T(\Phi - \lambda_1\Psi)\boldsymbol{y} \\
&= 2\boldsymbol{y}^T\left(\Phi\sum_{i=1}^n \alpha_i\boldsymbol{v}_i - \lambda_1\Psi\sum_{i=1}^n \alpha_i\boldsymbol{v}_i\right) \\
&= 2\boldsymbol{y}^T\left(\Psi\sum_{i=1}^n \lambda_i\alpha_i\boldsymbol{v}_i - \Psi\sum_{i=1}^n \lambda_1\alpha_i\boldsymbol{v}_i\right) \\
&= 2\left(\sum_{i=1}^n \alpha_i\boldsymbol{v}_i\right)\Psi\left(\sum_{i=1}^n \alpha_i(\lambda_i - \lambda_1)\boldsymbol{v}_i\right) \\
&= 2\sum_{i=1}^n\sum_{j=1}^n \alpha_i^2(\lambda_j - \lambda_1)\boldsymbol{v}_i^T\Psi\boldsymbol{v}_j \\
&= 2\sum_{i=1}^n \alpha_i^2(\lambda_i - \lambda_1)\boldsymbol{v}_i^T\Psi\boldsymbol{v}_i
\end{aligned}$$

$$= 2 \sum_{i=1}^{n} \alpha_i^2 (\lambda_i - \lambda_1) < 0$$

Therefore, $\boldsymbol{v}_1$ satisfies a necessary condition to be a local maxima.

We now show that no other point stationary points of $L(\boldsymbol{x}, \lambda)$ except for another generalized eigenvector corresponding the largest eigenvalue, satisfies the necessary condition to be a local maximum. Then $H(\boldsymbol{v}_i, \lambda_i) = 2(\Phi - \lambda_i \Psi)$ and let $\boldsymbol{y} = \boldsymbol{v}_1$. Since $\boldsymbol{v}_i^T \Psi \boldsymbol{v}_1 = 0$, $\boldsymbol{y} \in P(\boldsymbol{v}_i)$.

$$\begin{aligned}
\boldsymbol{y}^T H(\boldsymbol{v}_i, \lambda_i) \boldsymbol{y} = 2\boldsymbol{v}_1^T (\Phi - \lambda_i \Psi) \boldsymbol{v}_1 &= 2\boldsymbol{v}_1^T (\Phi \boldsymbol{v}_1 - \lambda_i \Psi \boldsymbol{v}_1) \\
&= 2\boldsymbol{v}_1^T (\lambda_1 \Psi \boldsymbol{v}_1 - \lambda_i \Psi \boldsymbol{v}_1) \\
&= 2(\lambda_1 - \lambda_i) \boldsymbol{v}_1^T \Psi \boldsymbol{v}_1 > 0
\end{aligned}$$

Therefore $(\boldsymbol{v}_i, \lambda_i)$ does not satisfy the conditions for being a local maximum.

Thus, only eigenvectors corresponding to the largest eigenvalue satisfy the necessary conditions. Also, for any eigenvector, $\boldsymbol{v}_1$, corresponding to the largest generalized eigenvalue of (Φ, Ψ) that also satisfies $g(\boldsymbol{v}_1) = 0$, $f(\boldsymbol{v}_1) = \lambda$. Thus, all points that satisfy the necessary conditions to be a local maxima have the same value λ_1. Thus, the global maxima is λ_1 and it is achieved by an eigenvector corresponding to the largest generalized eigenvalue. □

References

[1] C. E. Shannon, "Communication in the presence of noise," *Proceedings of the IEEE*, vol. 72, no. 9, pp. 1192 – 1201, 1948.

[2] H. Nyquist, "Certain topics in telegraph transmission theory," *Proceedings of the IEEE*, vol. 90, pp. 280 –305, 2002.

[3] E. T. Whittaker, "On the functions which are represented by the expansion of interpolating theory," *Proc. Roy. Soc. Edinburgh*, vol. 35, pp. 181–194, 1915.

[4] V. A. Kotelnikov, "On transmission capacity of ether and wire in electrocommunications," *R.K.K.A., Moscow*, 1933.

[5] E. Parzen, "Simple proof and some extensions of sampling theorems," Stanford University, Palo Alto, CA, Tech. Rep., 1956.

[6] L. Fogel, "A note on the sampling theorem," *IRE Trans.*, vol. 1, pp. 47–48, 1955.

[7] W. D. Montgomery, "K-order sampling of n-dimensional bandlimited functions," *Int. J. Contr.* , vol. 1, pp. 7–12, 1965.

[8] A. Kohlenberg, "Exact interpolation of band-limited functions," *J. Appl. Physics*, vol. 24, pp. 1432–1436, 1953.

[9] A. J. Jerri, "The shannon sampling theorem - its various extensions and applications: A tutorial review," *Proceedings of the IEEE*, vol. 65, no. 11, pp. 1565 – 1596, nov. 1977.

[10] R. J. Marks, Ed., *Advanced topics in shannon sampling and interpolation theory.* Springer-Verlag, 1992.

[11] A. Papoulis, *Signal Analysis.* McGraw-Hill, 1977.

[12] A. I. Zayed, *Advances in Shannon's Sampling Theory.* CRC, 1993.

[13] G. Dahlquist and A. Bjorck, *Numerical Methods.* Prentice-Hall, 1974.

[14] D. R. Kincaid and E. W. Cheney, *Numerical Analysis: Mathematics of Scientfic Computing.* American Mathematical Society, 2000.

[15] A. V. Balakrishnan, "A note on the sampling principle for continuous signal," *IRE Trans. Inform. Theory*, vol. 3, pp. 143–146, 1957.

[16] J. Sacks and D. Ylvisaker, "Designs for regression problems with correlated errors," *The Annals of Mathematical Statistics*, vol. 37, no. 1, pp. 66–89, 1966.

[17] ——, "Statistical design and integral approximations," *Canadian Mathematical Congress*, 1970.

[18] J. Hajek and G. Kimeldorf, "Regression designs in autoregressive stochastic processes," *The Annals of Statistics*, vol. 2, pp. 520–527, 1974.

[19] G. Wahba, "On the regression design problem of sacks and ylvisaker," *The Annals of Mathematical Statistics*, vol. 42, pp. 1035–1053, 1971.

[20] Y. Su and S. Cambanis, "Sampling designs for estimation of a random process," *Stochastic Processes and their Applications*, vol. 46, no. 1, pp. 47 – 89, 1993.

[21] C. A. Micchelli and W. Wahba, "Design problems for optimal surface interpolation," *Approximation Theory and Applications*, pp. 329–348, 1981.

[22] D. E. Boyce, A. Farhi, and R. Weischedel, *Optimal Subset Selection.* Springer-Verlag, 1974.

[23] A. Miller, *Subset Selection in Regression.* Chapman and Hall, 1990.

[24] A. Das and D. Kempe, "Algorithms for subset selection in linear regression," in *ACM symposium on Theory of computing*, ser. STOC '08, New York, NY, USA, 2008.

[25] N. Cressie, "Statistics for spatial data," *Terra-Nova*, pp. 613–617, 1970.

[26] C.-W. Ko, J. Lee, and M. Queyranne, "An exact algorithm for maximum entropy sampling," *Operations Research*, vol. 43, no. 4, pp. 684–691, 1995.

[27] C. Guestrin, A. Krause, and A. P. Singh, "Near-optimal sensor placements in gaussian processes," in *Proceedings of the 22nd international conference on Machine learning.* New York, NY, USA: ACM, 2005, pp. 265–272.

[28] S. T. McCormick, *Submodular Function Minimization.* Elsevier, 2006.

[29] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions," *Mathematical Programming*, vol. 14, pp. 265–294, 1978.

[30] D. Marco, E. Duarte-Melo, M. Liu, and D. L. Neuhoff, "On the many-to-one transport capacity of a dense wireless sensor network and the compressibility of its data," in *IPSN.* Berlin, Heidelberg: Springer-Verlag, 2003, pp. 1–16. [Online]. Available: http://portal.acm.org/citation.cfm?id=1765991.1765993

[31] E. J. Duarte-Melo and M. Liu, "Data-gathering wireless sensor networks: organization and capacity," *Computer Networks*, vol. 43, no. 4, pp. 519 – 537, 2003.

[32] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *International Conference on System Sciences*, 2000.

[33] S. Lindsey, C. Raghavendra, and K. M. Sivalingam, “Data gathering algorithms in sensor networks using energy metrics,” *IEEE Transactions on Parallel and Distributed Systems,*, vol. 13, no. 9, pp. 924 – 935, 2002.

[34] X. Zixiang, A. D. Liveris, and S. Cheng, “Distributed source coding for sensor networks,” *IEEE Signal Processing Magazine*, vol. 21, no. 5, pp. 80 – 94, 2004.

[35] A. Kashyap, L. A. Lastras-Montano, C. Xia, and Z. Liu, “Distributed source coding in dense sensor networks,” in *DCC*, 2005, pp. 13 – 22.

[36] G. E. Uhlenbeck and L. S. Ornstein, “On the theory of the brownian motion,” *Phys. Rev.* , vol. 36, no. 5, pp. 823–841, Sep 1930.

[37] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, Eds., *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide.* SIAM, Philadelphia, 2000.

[38] T. K. Moon and W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processing.* Prentice Hall, 2000.

CHAPTER V

Conclusions and Future Work

In this final chapter, we provide a summary of the contributions of this dissertation and avenues for future work.

5.1 Summary

In this dissertation, we investigated three problems that wireless ad hoc networks face. The first problem addressed is that of throughput scaling in random access wireless networks, while the second two were more specific to field gathering wireless sensor networks. Chapter II examined the problem of throughput scaling in random access wireless ad hoc networks. The networks under consideration were extended networks in which both the number of nodes and the area of the network increase at the same rate. This ensures that the density of nodes remains constant. Each node in the network acts as a source and wants to communicated to a destination that is located at a random location. Franceschetti et al. recently improved the throughput scaling to $\Omega_p\left(\frac{1}{\sqrt{n}}\right)$ from $\Omega_p\left(\frac{1}{\sqrt{n \ln n}}\right)$ as achieved by Gupta-Kumar. By constructing a routing and scheduling strategy, we showed that the increase in throughput scaling is due to percolation based route construction. Percolation allows us to construct routes from each source to its destination that have all but two of their hops have lengths $O(1)$. Shorter hop lengths allow more users to transmit simultaneously, while still allowing each of the nodes to transmit at a rate that does not depend on the number of users in the network. This helps in improving the throughput. The benefit of using the capacity-based link bit-potential model instead of a threshold-based model is that it allows the power at each node to remain constant even as the network size increases. However, under the capacity bit-potential model the nodes have to receive data at SINR that decreases to zero with increasing network size. Thus, there is a trade-off between using a powerful transmitter or a powerful receiver. There is no benefit obtained from using a highway based routing strategy of Franceschetti et al, as compared to the semi-straight line routing used by Gupta-Kumar.

Chapter III examined the trade-off between compression and robustness of distributed coding schemes in wireless sensor networks. We provided a rigorous theoretical formulation of the problem and measured performance of the schemes in terms of average rate and loss factor. It was shown that traditional Slepian-Wolf coding in wireless sensor networks suffers from a catastrophic loss if even one of the sensors fails. In order to use distributed coding in scenarios where nodes may fail, an extension to the original Slepian-Wolf theorem was provided. The extended Slepian-Wolf theorem allows for designs of decoders that can decode from a subset of the encoders that satisfy rate conditions given in the theorem. We then proposed and analyzed several different flexible distributed coding schemes. The schemes, while working at low average rates, are very robust to encoder failures. The performance of these schemes was evaluated for a one-dimensional sensor network measuring a stationary Gauss-Markov field. The evaluation shows that flexible distributed coding schemes can achieve the smallest possible loss factor, equaling the probability of node failure, for rates significantly smaller than independent coding.

Chapter IV investigated the problem of data collection in field gathering wireless sensor networks. The first part of the chapter examines the sensor placement problem.

In this problem, the senors transmit the data at their locations to the collector without using relays. The collector on receiving data from all nodes then reconstructs the field in the entire network region. It was shown that for a sensor network measuring a one-dimensional Markov field with exponential correlation, the optimal placement of sensors under the mean squared error criteria is uniform. In the second part of the chapter, a data gathering algorithm for sensor network was introduced. In the algorithm each sensor linearly combines its own data with the data it relays for other nodes. The collector on receiving the fused data makes an estimate of the field at the current time. After optimizing the process of data fusion, it was observed that the optimal density of nodes increases with increasing temporal correlation and decreases with increasing spatial correlation. This suggests that there is an optimal density when gathering data from sensors under a communication constraints.

5.2 Future Work

The research described in this dissertation lays the ground for a multitude of future problems for study.

The foremost open question in the throughput scaling problem in random access communication networks is the evaluation of the pre-constants that multiply the order results. As a first step, it would also be interesting to investigate how quickly does throughput converge to the formulas given in Chapter II.

For the reliability problem, studied in Chapter III, an interesting problem would be to find the achievable loss-rate function and the achievable rate-loss region. We also envision a fully flexible scheme without an underlying structure. In this scheme, the decoder on receiving encoded indices from some of the encoder, checks which of the nodes it can decode based on which subset's satisfy the extended Slepian-Wolf conditions. Finally, constructing practical distributed codes for flexible schemes is an open problem, too.

For sensor placement and data gathering in wireless sensor networks, an interesting problem to pursue would be generalizing the one-dimensional placement result to non-Markov processes and the extension of the placement result to two dimensions. As a first step in extending the result to two dimensions, it may be helpful to look at the asymptotic case. For the data gathering problem, a more rigorous theoretical analysis of how the density of nodes depends on the spatial and temporal correlation is still needed. Also, of interest would be extending the current algorithm to case of periodic schedules. Once again, extensions to two dimensions is still open.