OPTIMAL TOPOLOGIES AND ALGORITHMS FOR MINIMIZING DATA RETRANSMISSIONS IN WIRELESS NETWORKS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Milen Valentinov Nikolov

January 2015

OPTIMAL TOPOLOGIES AND ALGORITHMS FOR MINIMIZING DATA RETRANSMISSIONS IN WIRELESS NETWORKS

Milen Valentinov Nikolov, Ph.D.
Cornell University 2015

Wireless networks comprise the majority of devices within the growing edge of the global communication system. Performance metrics determining the successful application of wireless networks in that setting are goodput, latency and network lifetime. Overhead retransmissions due to redundant data transfer, inefficient transmissions, low link quality, and suboptimal network layer protocols affect negatively these three metrics. Designing wireless networks to minimize the overhead retransmissions encompasses three network levels: the data, structural and procedural levels. Encoded sensing (ES) is a "data-aware" scheme that shapes the network *structural level* to account for correlations across data sources and common data across groups of nodes. Via new encoding algorithms, ES achieves substantial reduction of the transmissions required to convey a message to a sink node. A few beneficial properties for network hardware and design, based on sparsity of ES signals, are also discussed. The *structural level* is further augmented by the placement of relay nodes to minimize the overhead retransmissions in the network due to low quality and heavily loaded links. Finally, the Time Sequence Scheme operates on the network *procedural* level, allowing for broadcast of messages reaching all network nodes, while minimizing redundant broadcast retransmissions. Explicitly minimizing the number of retransmissions at each of the three network levels impacts beneficially performance as shown by analysis and full network stack simulations.

BIOGRAPHICAL SKETCH

Milen Nikolov received a B.Sc. degree in Computer Science and a B.Sc. degree in Mathematics from The State University of New York: College at Brockport, both in 2009. From 2009 until 2015 he pursued research in the department of Electrical and Computer Engineering at Cornell University, Ithaca, NY, where he completed his M.Sc. degree in 2014 and his Ph.D. degree in 2015, under the advisement of Prof. Zygmunt J. Haas. Milen Nikolov's interests are in the areas of wireless communications and information networks. He has been studying algorithms and processes over networks with dynamic topologies, relay networks topology control, and collaborative communication in information networks. His research has been published in the IEEE Transactions on Mobile Computing and a number of international conferences proceedings.

Milen Nikolov has received awards and honors from different venues and organizations for his work and research. With Prof. Zygmunt J. Haas, he was awarded the Best Paper Award at the 2012 ACM MobiCom International Workshop on Mission-Oriented Wireless Sensor Networking for developing and studying Encoded Sensing for Wireless Networks. He also received the State University of New York Chancellor's Award for Student Excellence, in 2009, and the Symbian Ltd. International Research Essay Contest award for Breakthrough Technology Improvements, in 2008.

To the patience of all my family and friends. There are a number of people without whose support (despite my occasional and unintentional neglect for communication) I would not have been able to stand at the completion point of this endeavor.

To my parents Violeta Nikolova and Valentin Nikolov: a major portion of the work in this thesis is your doing. It is read between the lines; written guidance dense and white in the margins; invisible to others, it is now transparent to me.

To my brother Georgi and my sister Dani: our conversations and experiences, across all the different time-zones and places of the world, frequently inspired what I wanted to do and directed me when in doubt.

To Ivo, Krisi and Radoslav (notice the alphabetical order): the last 4-5 years in Ithaca went by almost too quickly thanks to you.

ACKNOWLEDGEMENTS

I would like to especially thank my adviser Prof. Zygmunt Haas. My work has been made both possible and enjoyable through his patience, guidance, support, and expertise. The many meetings we have had frequently led to new ideas and shaped my understanding of science and research. I am very fortunate to have Prof. Al Molnar and Prof. Joe Halpern as my committee members. I would like to thank them for their flexibility, time and agreeing to serve on my committee. I also would like to thank Prof. Kevin Tang for sitting on my exams, the stimulating questions and meetings.

My thanks go to Scott Coldren and Daniel Richter of the ECE department's administration. Their quick replies and timely reminders were invaluable.

I would like to thank the department of Electrical and Computer Engineering for providing me with a great scholarly environment.

My research has been supported by the NSF (respectively grant numbers ANI-0329905, CNS-0626751, CNS-1040689, and ECCS-1308208) and AFOSR (contract number FA9550-09-1-0121/Z806001). I would like to thank these organizations for their continual support.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

"But they are useless. They can only give you answers."

–Pablo Picasso on computers

1.1 The Expanding Edge

In 2016, the projected total world-wide internet traffic will exceed 1 zetabyte per year, for the first time [1]. While this is still far away from the 10^{90} total number of bits the universe could register were it an ultimate computer [2], in the context of human records this number describes approximately the content of 174 newspapers received per person every day, per year [3].

Again in 2016, the projected amount of traffic generated by wireless devices for the first time will surpass the traffic generated by wired devices [4]. More than 54% of the 1 zetabyte per year will be generated by tablets, smart machine-to-machine (M2M) devices, sensors, and wearable electronics residing at the edge of the network. And the edge is expanding: the Compound Annual Growth Rate (CAGR) of the global communication system's edge (estimated at 10.7%) is growing faster than the world population (CAGR 1.1%) and the number of Internet users (CAGR 9.2%) *combined* [4]. This shows that the average number of interconnected devices per person will be more than one. A typical household in the U.S. will own a sizeable, wireless communication network comprising dozens of devices.

The projected numbers in [4] also help outline the nature of devices in such wireless communication networks. Characterized by fast deployment and mobility, these devices are energy-limited via battery power supply and communicate with the rest of the network via range-limited links. As discussed in this thesis, the amount of transmissions on the wireless links determines the cost, performance, and lifetime of a network. The efficiency of state-of-the-art wireless networks technology and deployment can be substantially improved in various domains (sensor networks, cyber-physical systems, health monitoring medical wearable devices, etc.), significantly reducing the number of transmissions and overhead (re)transmissions required to carry more than 500 exabytes of data annually over the next few years.

This thesis outlines three network levels and their functional relationship to the number of generated radio transmissions within a wireless network: the data, the structural, and the procedural network levels. The data level shapes the optimal construction of the structural level; in turn, the structural level guides the design of the optimal algorithms and protocols residing at the procedural level. Fig. 1.1 shows an instance of the three-level decomposition for a sample wireless sensor network. To optimize the number of (re)transmissions and energy consumption in the network, three schemes are suggested in this work: Encoded Sensing, RePlace, and the Time Sequence scheme. Each scheme is closely coupled with a distinct network level. Encoded Sensing exploits the properties of the data level and coordinates network devices transmissions to form collaborative, sparse codewords accurately representing the source data, yet requiring minimal number of transmissions. RePlace optimizes the properties of the structural level by placing relay nodes at strategic positions in the network, increasing links' reliability depending on links' traffic loads and thus

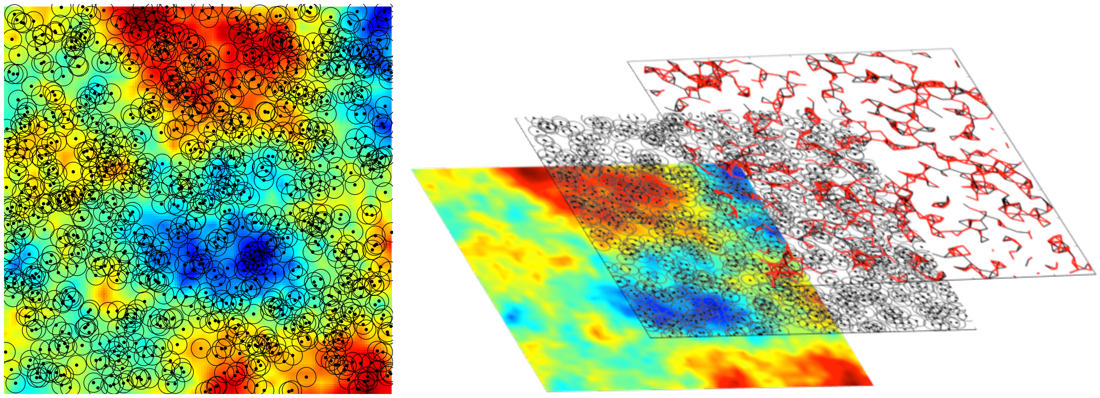


Figure 1.1: Left: a sample wireless sensor network measuring a physical phenomenon (e.g. temperature). Nodes are in black and their sensing ranges are homogeneous, as shown. Right: the decomposition of the network system into three levels - data (bottom), structural (middle), and procedural (top).

reducing packet loss and overhead retransmissions in the network. The Time Sequence Scheme resides at the procedural level and orchestrates the fundamental broadcast network procedure, so that a minimal number of nodes are required to transmit in order to convey a message from a source to all other nodes in the network.

The data level comprises the description of traffic sources (e.g. phenomenon to be sensed by a wireless sensor network, the traffic patterns input to an intelligent home network, etc.). The data level is rarely under the control of the network engineer. The network engineer can react to the nature of the data level by designing properly the structural and procedural network levels. Chapter 2 of this thesis examines how the network engineer could exploit knowledge of the data level in order to achieve network structure which minimizes the number of signals required to convey a message - accurately describing the source data - further up the network. The provided solution suggests new data encoding schemes and drives the optimal construction of a wireless network topology into specifically designed clusters of “source” nodes pushing information to a “head” node, so that the total number of signals sent by the source nodes is minimized. The resulting scheme reduces at least two-fold the amount of signaling compared to typical state-of-the-art wireless network schemes.

Chapter 3 discusses the placement of relay nodes so that the number of retransmissions in the resulting network is minimized. For instance, if the “head” nodes from chapter 2 are too far apart, the quality of the links on the paths connecting them may be weak, requiring the addition of relay nodes to the network. The relay nodes do not interact with the network data layer. They are not sources of traffic and only retransmit the messages received by other nodes, such as the “head” nodes. The *RePlace* algorithm outlined in chapter 3, finds the placement of relay nodes, minimizing and even eliminating transmissions and overhead re-transmissions, depending on the available number of relay nodes.

Given an optimized structural level, the design of a number of algorithms and protocols residing in the procedural network level can be tailored to minimize the number of transmissions in the network still further. Chapter 4, studies one of the fundamental network procedures: broadcast. Broadcast allows any network node to send a message reaching all other network nodes. In some cases, broadcast may be the sole feasible data propagation mechanism in wireless networks. More importantly however, crucial network protocols (e.g. routing) utilize broadcast for link state updates and discovering paths to nodes. Efficient broadcast mechanisms satisfying a certain set of desiderata and minimizing the number of (re)transmitting nodes may determine the successful deployment of a network application. The Time Sequence Scheme described in chapter 4 designed with the explicit goal of transmissions optimization, provides a fresh perspective on broadcast and leads to network performance significantly surpassing existing schemes in the technical literature.

1.2 Why Data Retransmissions?

There are different network parameters that network designers can strive to optimize at different layers: interference, congestion, load balancing, network survivability, etc.. In almost all of these cases the goal is to provide network quality of service (QoS) that satisfies the demands of a given network application, in a best effort fashion. In many cases network QoS can be captured by two metrics: latency and goodput.

Latency is typically defined as the time elapsed between a network source node sending a packet and a destination node receiving that packet in full.

Goodput is the amount of application relevant information (in bits) that is received at a destination node every second. This measure excludes the amount of lower layer protocols' overhead, packet retransmissions due to collision, etc..

Another metric, *network lifetime*, is becoming increasingly more relevant in the context of battery powered devices populating the Internet of Things (IoT), M2M, and Wireless Sensor Networks (WSN), the latter being deployed monitoring various physical processes, ranging from measuring soil moisture (e.g. [5]) for precision agriculture to assessing power consumption in buildings (e.g. [6]). In these scenarios, battery replacement is sometimes costly (industrial M2M networks cannot be shut-downed frequently for device maintenance) and is not always feasible (e.g. remotely deployed sensor networks). Although different authors assume a variety of network lifetime definitions varying with respect to the network applications (the reader may refer to [7] for overview), typically these definitions capture the amount of time in which the network devices are capable of satisfying QoS constraints. The network lifetime depends crucially on the power drained for devices' operation. It is well-known (e.g. [8], [9], [10]) that radio operation is one of the main energy depletion source for wireless devices. $60-65\%$ of the WSN nodes' energy is consumed by radio communication (this varies little with platforms); about 70% of WiFi enabled devices' energy is consumed by the radio operation, as well.

Examining the operation of various wireless network protocols, an interesting pattern emerges. While a number of variables affect network performance as captured by latency, goodput and network lifetime, *the number of transmissions* affect significantly all three of those metrics.

More specifically, for instance, on the MAC layer, per the IEEE 802.11 MAC protocol, the transmitter does not receive an acknowledgement by the receiver if a packet is dropped at the receiver. In turn the transmitter retries sending the packet. The number of such retries/retransmissions effectively leads to loss in the goodput of the network. Due to the retransmission backoff mechanism in the IEEE 802.11 MAC protocol, ceteris paribus, each retransmission until the source receives acknowledgement leads to increasingly larger packet delay. Furthermore, each transmission drains the energy of the devices. Therefore one would like to ensure that communication links are as reliable as possible and have as low communication cost as possible. As discussed in the following chapters, minimizing the number of overhead retransmissions is a function of the network structure: network engineers can design/control network topology to reduce these retransmissions

At higher network layers, larger number of (re)transmissions due to protocol design again leads to reduction of goodput, increase of latency and decrease of network lifetime. For example, “the broadcast storm” ([11]) problem results from inefficient flooding protocols propagating link status updates (or other data) through the network. In this case, the number of nodes retransmitting a broadcast message is too large causing contention, packet collisions, and ever more further retransmissions.

Therefore, optimizing network structure (e.g. via topology control) and network procedures (e.g. broadcast) to reduce the number of network transmissions enhances significantly network performance across different metrics. The following sections provide an overview of the prior work aimed towards minimizing the number of transmissions in wireless networks exploiting the characteristics of each of the three network levels.

1.3 The Data Level

Informally, in the context of data networks, a set of S_t source nodes may be thought of observing s_t pieces of data at time t. The data pieces contain information of I_s [bits]. Upon receiving I_s, a set of D_t destination nodes would change their states. The function of a data network is to move the amount of information I_r [bits] required for the *proper* change of states at the D_t destination nodes in *timely* and *reliable* fashion, where “proper”, “timely” and “reliable” depend on the network application requirements. [1]

In many cases the s_t pieces of data contain redundant information. For instance, in a wireless sensor network (WSN) a set of S_t nodes may be measuring the values of some phenomenon at a set of s_t points in space. The resulting s_t measurements may contain redundant data due to the spatial correlations inherent in many phenomena (e.g. temperature). The bits of information I_s obtained from all s_t measurements does not require the transmission of all s_t packets, each packet encoding a separate measurement.

The S_t nodes could *compress* the s_t packets to be sent. Alternatively, one could ask if the set of S_t source nodes itself can be pruned. In effect, fewer nodes, S'_t, *sample* the information in the s_t data packets and only the resulting subset of s'_t packets are transmitted. The two problems of efficient *compression* and *sampling* depending on the properties of the data sources and network communication channels have been central in the field of communication networks, and in the next subsections, we review some of the work related to the data network level.

1.3.1 Data Compression

One of the first results on network compression from the 1970s characterizes Distributed Source Coding (DSC). DSC is a network data compression technique leveraging results based on the Slepian-Wolf source coding theorem [12]. Namely, correlated data sources (e.g. the s_t data packets) can be compressed

[1] Notice that $I_r \leq I_s$. I_r may be less than I_s, for instance, if the source nodes "know" that the destination nodes already have access to portion of the information encoded in I_s. I_r could even be 0 if the source nodes "know" that the destination nodes already have access to all of the information in the I_s bits; or, that the receipt of the I_s bits would not change the state of the destination nodes. In this work, we assume that $I_s = I_r$ bits need to be delivered to the destination nodes.

separately - without communication between the source nodes - and without loss to the level of the joint entropy of the sources. This can be done as long as proper statistical correlation information is available at each of the source nodes.

For instance, in the case of WSNs different approaches for distributed source coding have been suggested in the recent technical literature utilizing available spatial correlation models for a number of physical phenomena. The authors of [13] and [14] leverage results based on the Slepian-Wolf source-coding theorem. According to a pre-specified scheme, each sensor node in the network transmits fewer bits the more correlated its measurement is to reference nodes. A somewhat different utilization of DSC is suggested in [15], where the authors extend DSC to Distributed Source Coding Using Syndromes (DISCUS); the main ideas however remains similar and are outlined below.

For simplicity consider the single-hop variation of the algorithms in [13] and [14]. Given a neighborhood radius, r_i, around a representative node i, the distributed source coding algorithm constructs an ordered sequence C_i of the nodes that are within distance r_i of i. The sequence can be ordered based on the distance between the sink and each node in the sequence (including i). (If the nodes are approximately equidistant to the sink the sequence can be ordered based on nodes' IDs.)

Next, in the order of the sequence, rates (in bits) are allocated as follows: $R_1(H(X_1))$ to the first node in the sequence, $R_2(H(X_2|X_1))$ to the second node in the sequence, and so on until the last node transmits with lowest rate: $R_N(H(X_N|X_N, X_{N-1}, ..., X_1))$, assuming $|C_i| = S_t = N$. Note, that while only local information is required, all nodes within distance r_i of i need to know all distances between the N nodes in C_i. Given a specific spatial correlation model, the distance information is required to compute the conditional entropies determining each node's rate.

Here, the phenomenon's statistical model is assumed to be a Gaussian random field whose instances in the plane are Joint Random Gaussian Variables (JGRV) with Power Exponential correlation model. In this case, one can find, following [16], that the conditional entropy $H(X_j|S \subset C_i)$ is approximated using the conditional differential entropy as follows

$$h(X_j|S \subset C_i) = 0.5log\left((2\pi e)^{|C_i|-|S|}\,\frac{det(\mathbf{K})}{det(\mathbf{K[S]})}\right)$$

where $\mathbf{K}$ is the covariance matrix of the JGRV observed by the nodes in C_i. $\mathbf{K}[S]$ is the covariance matrix of the JGRV observed by the subset, S, of nodes in C_i that node j conditions on, depending on j's order in the sequence C_i. The covariance matrix is computed using the given Power Exponential correlation model and the known distances across the nodes.

Hence, instead of sending s_t packets containing redundant information, the S_t nodes are grouped in a sequence C_i with respect to some reference node i and each of them only sends information conditional on the information sent by the rest of the nodes in the sequence. As noted in [13] and [14], a number of applications ([17], [18], [19] and [20] among others) in sensor and low power wireless networks are suitable to utilize a similar DSC scheme.

Despite the rates allocated by DSC account for nodes observations' correlation, still the total rate equals at least the joint entropy of the observed measurements. Proper *sampling* can be utilized in conjunction with compression to reduce the amount of information sent by source node even further. The trade-off here is that the lower the number of data samples sent the greater the distortion of the received information at the sink.

1.3.2 Data Sampling

Data sampling across a set of S_t source node is most intuitive in the context of WSNs, though the sampling approaches considered below can be generalized to different applications where a set of nodes have access to data packets containing redundant/correlated information.

In a wireless sensor network setting, for example, a scheme is introduced in [21] eliminating the transmission of highly correlated measurements across the nodes in S_t. Notice that in contrast to DSC discussed above where all nodes transmitted, albeit potentially small amount of information, here only a subset of representative nodes transmits the entirety of their measurements. As shown in [21], the elimination of certain highly correlated messages would not distort (within a bound) the overall phenomenon estimate at the sink. Thus network energy is conserved via appropriate spatial sampling. Fig. 1.2 illustrates the representative nodes approach in the context of WSNs. The outcome of the scheme is that within a group of S_t nodes measuring the same (or very similar) values, only a single representative node transmits an entire measurement value at every time instance t. The work in [21] is motivated by the results in [22] and [23] among others investigating the relation between the number of samples (spatial, spatial-temporal respectively) and the achieved distortion of the sink. The latter studies demonstrate that as the network density increases in WSN the number of samples, as a percentage of the total number of nodes, may decrease substantially without significant phenomenon estimate distortion increase at the sink. A similar redundant node elimination approach has been suggested in [24] where the authors determine the set of representative nodes based on multiple distinct set covers. Here the goal is to cover a discrete set of

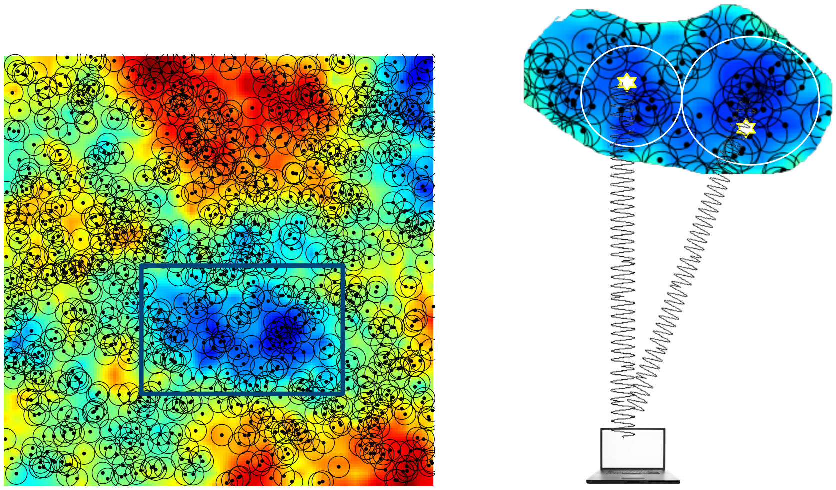


Figure 1.2: A number of network areas contain a large number of nodes with spatially correlated measurements. One such area is shown in blue. The area contains two clusters of nodes. The nodes in each cluster measure very correlated measurements that can be represented by the transmission of only one representative node per cluster.

targets in contrast to accurately represent a continuous physical phenomenon. At any given time only the nodes in one of the set covers is active. Each set cover comprises the representative sensor nodes accurately depicting the set of observed targets.

A more recent development of sampling compression schemes utilizes the theory of compressive sensing ([25], [26], and [27]). The theory and algorithms offered by compressive sensing depend on the structure of the data describing a given phenomenon over time and space. Often physical phenomena can be accurately described by signals possessing sparse representation in a certain basis. In the context of wireless sensor networks the signals may have sparse representation over time at a given sensor node (intra-signal sparse structure) and/or over space across the measurements of different sensor nodes at any fixed time (inter-signal sparse structure).

Intra-signal structure has been exploited beginning with the work of Candes and Wakin ([28]) so that the measurements collected over *time* by any given node are represented by substantially less data. For instance, suppose the phenomenon source can be represented as a K-sparse signal over a period of n timeslots. In the context of WSN, instead of sampling the phenomenon at each timeslot and transmitting a message for a total of n messages, each node may sample only at $O(Klog(n))$ timeslots. For example, in a very recent work, the authors of [29] apply intra-signal compressive sensing techniques to the problem of in-situ soil moisture measuring. Various other schemes based on compressive sensing rely on inherent K-sparsity in phenomenon signal's structure. An approach to utilizing compressive sensing to account for inter-signal sparsity due to spatial correlation is presented in [30]. The authors utilize specifically selected clusters of spatially correlated nodes and apply compressive sensing locally within these clusters. While in the intra-signal compressive sensing recent works relax the assumption that the observed signal is exactly K-sparse over time (e.g. [31], [32]), in the inter-signal setting [30] require that the observed phenomenon signals at a set of locations have exactly K-sparse representation (in some basis); furthermore, the signals need to preserve spatial K-sparseness over time.

Notice that in themselves the compressive sensing-based temporal sampling methods are orthogonal to the techniques utilizing spatial correlation across nodes discussed above. Hence, a combination of spatial and temporal compression would be potentially beneficial. In a foundational study ([33]), the authors investigate distributed compressive sensing applied to a signal characterized by inter-signal K-sparsity, extending compressive sensing to distributed source coding. The authors assume exact K-sparse representation of the phenomenon over time.

The above approaches illustrate that properties of the data at the source nodes directly influence the *amount* of transmissions in the network. As we will discuss in chapter 2, the data level characteristics may also determine *how* to design optimal transmission strategies for that data. E.g. cooperative transmissions across nodes with access to “common data” reduce the required transmission power for each bit, while achieving the same signal to noise ratio at the receiver via beam-forming (e.g. [34] and [35]); also, sparsity of the data at the source nodes may be exploited to modulate nodes’ transmissions, so that the resulting transmitted signal contains only a low number of data’s random projections while the destination node can still recover the data using wireless compressive sensing (as suggested in [36]). The approach of Encoded Sensing presented in chapter 2 improves on these methods by introducing a novel collaborative coding scheme, based on “common data” and “approximately common data”. We do not require sparsity in the data at the source nodes, and yet are able to exploit the benefits of compressive sensing at the physical layer due to the properties of the constructed collaborative codebook.

1.4 The Structural Level

A number of important network performance properties are determined by network’s topology links’ states. Network’s structural level captures topological and links states characteristics. The structural level design reflects network’s connectivity, fault tolerance and links’ reliability.

1.4.1 Connectivity and Fault-tolerance

A network is connected if for any pair of nodes u and v in the network there exists a path of bi-directional reliable links that connects u and v. The reliability of each link connecting node i to node j on a given path is determined by the amount of noise and interference on that link and on the transmission power of the source node (e.g. i). It is well known (see for instance [37]) that the transmitted wireless signal deteriorates proportional to d_{ij}^{α}, where d_{ij} is the distance between nodes i and j and in various wireless networks $\alpha \in [2, 4]$. Hence, intuitively to increase the reliability of the links in the network and the probability that the network is connected at any given time, the wireless nodes should increase their transmission power. This strategy is not necessarily beneficial:

- First, increasing transmission power decreases network life time in the case of battery constrained devices.

- Second, increasing i's transmission power may lead to increase in interference across a different set of links.

1.4.1.1 Transmission Power Control

Finding the minimum level of transmission power, so that the network is still connected is one approach of controlling network topology and optimizing the network's structural level. Various algorithms have been suggested in the technical literature for power allocation and transmission radius adjustment to maintain connectivity, while maximizing network life. Li et al. ([38]) determine the minimum amount of power required to maintain a cone originating at source node i and spanning x degrees, so that a message sent by i is delivered successfully at least at one node inside the cone. The authors derive the optimal value for x, so that the network is connected. In contrast, the authors of [39] minimize the maximum transmit power across all nodes in the network by allocating transmission powers to the nodes following a Minimal Spanning Tree strategy. A strategy of incremental adjustment of transmission powers to minimize the maximum transmit power is also studied in [40]. Instead of minimizing the maximum assigned transmission power, a different approach could be the minimization of the *total* transmission power consumed in the network while maintaining connectivity (e.g. see [41]). The two different approaches to topology control optimize different life-time metrics ([42] and [7] provide more references to various lifetime models).

1.4.1.2 Relay Nodes Placement

In all of these cases the authors assumed that nodes perturb their transmission powers in order to optimize a given property (i.e. connectivity) of the network structural level. A different stream of work considers instead the addition of *relay nodes* to the network. These are assumed to be cheap devices that are not sources of the data in the network, and only re-transmit the packets reaching them. The authors of [43] and [44] investigate connectivity guarantees via relay node placement, instead of transmission power adjustment, and show that the relay placement problem is NP-hard in that setting. Various heuristic and approximation algorithms have been suggested to optimize the positions of the relay nodes. In many cases, a stronger condition on network connectivity is imposed: at least k paths need to exist between each pair of network nodes.

The latter condition provides survivability and fault-tolerance guarantees with the network deployment and relay nodes placement (for instance, if less than *k* nodes antennae fail it is guaranteed that the network will still be connected). The authors of [45], [46], [47], [48] and [49] and references therein provide various approximation schemes to the optimal relay placement, relying on operation research results regarding finding Steiner points in a network (e.g. [50]) and survivable network design problems (e.g. [51]).

1.4.1.3 Virtual Mobile Backbones

Another form of topology control specifically targeting Mobile Ad-hoc Networks (MANETs) relies on the construction of mobile virtual backbones. In these studies the authors assume that the network consists of two types of nodes: highly and less capable nodes. The network designer has control over the mobility patterns of the highly capable nodes. The task is to position the highly capable nodes w.r.t. the less capable nodes in order to improve certain network performance metrics. The first studies (e.g. [52] and [53]) describing the building and maintaining of such virtual mobile backbones did so in the context of routing (increased throughput) and network QoS (decreased latency). However, virtual mobile backbones has since been utilized (e.g. the reader may refer to [54]) to maintain network connectivity by solving the connected disk cover problem and minimizing the number of highly capable nodes.

1.4.2 Reliability and Communication Cost

Topology control can be used to achieve desired levels of network connectivity and path redundancy as discussed above, but it can also be utilized to increase network's links reliability and reduce overhead communication costs due to multiple packet re-transmissions. These are the metrics we concentrate on in chapter 3 of this thesis. In this context, network designers have resorted both to the placement of additional relay nodes in the network and to adjusting nodes positions. The latter technique relies either on controlling network nodes' mobility to create a virtual mobile backbone (similar to the one described in the previous subsection); or, altering the placement of the relay nodes via network re-configuration.

The approach of allocating mobile nodes to construct a virtual mobile network backbone has found popularity in works on mobile robots networks. Recently, in [55], the authors considered the optimal placement of M mobile relay nodes serving as a forwarding communication backbone to reduce network communication cost. Their approach improved on earlier works (e.g. [56] and [57]) of mobility-based topology control for communication cost optimization. In [56] the authors considered mobile relay nodes forwarding information across static source nodes. In [57], Goldenberg et al. provide a distributed algorithm controlling the mobility of the relay nodes towards the midpoints of links carrying traffic. In both works, the communication cost improvement in the resulting network has been measured in terms of energy saved, while maintaining fixed link reliability. An alternative equivalent communication cost improvement measure would be increasing link reliability while maintaining a fixed transmission energy budget. As shown in [55] the midpoints of certain loaded network links is not necessarily the best placement of the mobile relay nodes w.r.t. to the rest of the links in the network that are carrying traffic. Furthermore, the motion of the nodes (i.e. robot agents) depletes energy: the path of the mobile nodes needs to be optimized in order to conserve energy [55]. In the context of sensor networks, where nodes are often static and the adjustment

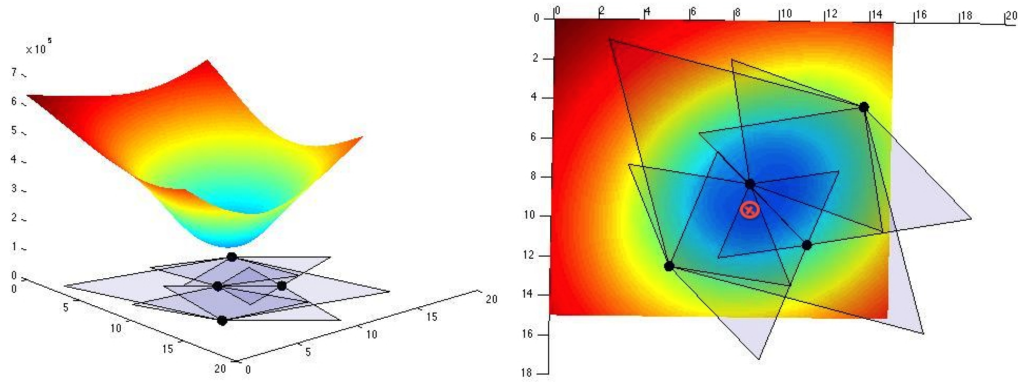


Figure 1.3: The communication cost function of a network on four fixed source nodes and one relay node. Each (x,y) position of the relay node corresponds to a communication cost value (z-axis), given fixed routing in the network (i.e. routing does not change as the relay nodes position changes). Notice that the communication cost function attains its minimum (marked with a cross) in the intersection of the specifically shaped zones, each zone associated with a distinct link. In this example, placing the relay node at the position marked by the cross also induces optimal routing in the network and minimizes globally the communication cost function. Notice that in general this need not be the case: the optimal routing may change as relay nodes' position changes and, vice versa, the optimal relay node's placement changes as the routing changes. Chapter 3 discusses in more detail the zone shape and the intersection selection for relay node placement.

of nodes' positions over time is not feasible, a set of additional relay nodes may be added to the network at arbitrarily picked and fixed positions, so that communication cost is minimized. The optimal placement of relay nodes have been studied in [58], [59], and [60]. In these cases however, the potential locations of relay nodes within the network are constrained to a discrete, finite, set of prespecified points on the plane. Chapter 2 describes in more detail the problem of relay placement in this setting. The RePlace algorithm for relay placement is studied along with the reformulation of the optimal relay placement problem that avoids inherent inefficiencies in current relay placement models (e.g.

in the works discussed above, traffic patterns and links loads are considered fixed as the relay nodes are placed). We observe that the communication cost function is convex when the routing in a network is fixed. We consider geometrical constraints on the optimal routing of traffic through the network in the presence of relay nodes. Fig. 1.3 illustrates an example of the communication cost function given a network of four fixed source nodes and one relay node placed in the intersection of specifically picked feasible zones for relay nodes placement. Chapter 3 motivates the zones' shape and the intersection selection for relay node placement, with the goal of minimizing the number of overhead retransmissions in the network and overall network communication cost.

1.5 The Procedural Level

Suppose we have established what, how much information to send, how to encode that information and what are the source nodes in the network at the data level. At the structural level, we have determined the positions of the nodes, or their mobility patterns (when applicable), have assigned transmission powers and placed communication relay nodes as needed to insure connectivity, increase reliability and reduce network communication cost. How and when should then information flow through the network so that links are not overloaded and congested, information is delivered from the source to the destination over the "best" possible paths according to some metric (for instance, number of retransmissions or energy consumption); which of the network nodes should forward information given a specified set of source node and a specified set of destination nodes? Questions of this flavor are addressed at the procedural network level.

The communication patterns in networks can be broken into three types: unicast, multicast and broadcast. Unicast comprises communication scenarios where typically a single source has information delivered to a single destination; multicast comprises communication patterns where a source node sends the same piece of information to a specific subset of the network nodes; and broadcast comprises patterns where a source node has to deliver the same piece of information to all nodes in the network.

1.5.1 Unicast

In wireless networks, links are often asymmetrical. If node A can communicate with node B that does not necessarily imply that node B can communicate with node A (the reader may refer to, for instance, [61] in the case of wireless ad hoc networks, [62] in the case of wireless sensor networks). Also, links' quality change over time (due to nodes' mobility, environmental changes, etc.); e.g. see [63] and [64] for link quality prediction in, respectively, mobile networks and vehicular ad-hoc networks (VANETs).

Due to the dynamics of wireless link quality and asymmetry traditional wireline network protocols such as OSPF ([65]) do not perform well[2]. The amount of link state updates an OSPF implementation would require to account for link dynamics is not feasible due to limited bandwidth, for instance.

A number of unicast routing algorithm have been suggested in the technical literature and have been deployed in the context of WANETs, MANETs,

[2] Medium Access Control (MAC) protocols for wireless networks have very different design and handle a host of communication scenarios that are not present at wireline link layers. Here, we do not focus on MAC layer performance, however the interested reader may refer to [66] and the more recent work in [67] for a survey on MAC layer protocols for wireless networks.

VANETs [34]. The goal of these algorithms is to deliver each payload message from the source to the destination with minimal control message overhead, while maintaining a (approximately) correct view of the network's structural level. A number of protocols that have been shown to attain these goals while delivering acceptable QoS levels. The protocols can be classified as reactive, pro-active and hybrid. Reactive protocols (such as Associativity Based Routing (ABR, [70]), Dynamic Source Routing (DSR, [71]), Ad-hoc On Demand Distance Vector Routing (AODV, [72]), Temporarily Ordered Routing Algorithm (TORA, [73]), etc.) rely on building a source-destination path, on-demand, when a packet needs to be sent. In contrast, pro-active protocols (Destination-Sequenced Distance-Vector (DSDV [74]), Optimized Link State Routing (OLSR [75]), the Cluster Based Routing Protocol (CBRP [76]), etc.) attempt to maintain an accurate view of the network structural state over time. Each node periodically updates a routing table from itself to all other nodes connected to the network. The first example of a hybrid routing protocol was proposed in 1997: the Zone Routing Protocol (ZRP [77]), where routing within specifically constructed zones is done pro-actively; routing in-between zones is done reactively. This approach leverages a balance between the control message overhead of on-demand protocols (a source-destination path is built only if a packet is sent outside a zone) and pro-active protocols (the state of the network is maintained only within a local zone, usually comprising a 2-hop neighborhood).

The operation of the above unicast protocols should account for network's structural and data levels as well. If the network structure supports multi-

[3] In the context of infrastructure wireless networks (e.g. cellular networks), routing is easier since mobile nodes are typically one hop away from the destination.

[4] We do not consider here networks with intermittent connectivity, which fall in the broader class of delay tolerant networks. The interested reader may refer to one of the first studies on the problem [68]; [69] suggests the use of epidemic routing for the first time, in that context.

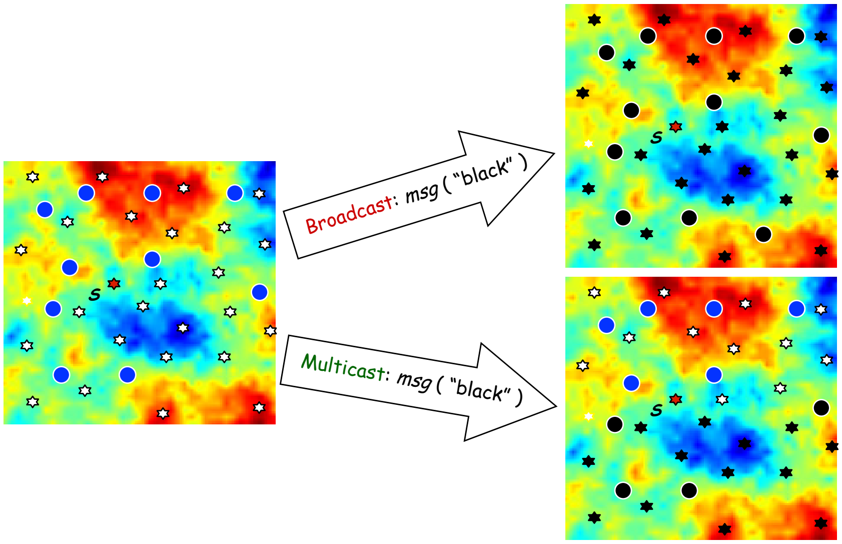


Figure 1.4: An example of sensor network with specified data and structural levels. Representative sensors (stars) at optimal sampling locations measure the values of a phenomenon following a Joint Gaussian Random Variable with Power Exponential spatial correlation model. The locations of representative sensors are chosen to minimize the number of representative sensors and transmissions required to convey accurately the phenomenon (data level). Relay nodes (blue circles) forward the measurements across representative nodes. Relay nodes' locations are optimized w.r.t. to the source data demand matrix and traffic loads in the network, in order to minimize communication cost and overhead transmissions due to dropped packets (structural level). Broadcast information flows from a source node (red) to all other nodes in the network. Multicast information flows from the source node to the subset of network nodes nodes south of it. Both multicast and broadcast are designed to meet network's QoS constraints and reduce redundant re-transmissions of packets.

ple paths between source and destination (each path having different bottleneck link capacity), an efficient routing protocol should attempt to balance the load across multiple paths over time and space. If only the minimum weight/shortest paths are utilized, links connecting nodes with high centrality would be heavily loaded increasing congestion and energy consumption due to dropped packets re-transmissions. Works along the lines of [78] and more recently [79] and [80] have investigated incorporating load balancing in wireless network routing protocols. In [79], the authors demonstrate that following optimal "curved" in lieu of short "straight" paths may decrease the maximum link traffic load in the network by 40%. As we see in chapter 3, traffic loads on links directly impacts the network structural level, for instance, in the context of optimal relay placement minimizing overhead transmissions in the network. In this work, we compute the optimal relay placement w.r.t. minimum weight source-destination path routing and investigate the resulting interplay between traffic loads and relays locations (chapter 3). However, a brief discussion on the impact of load balancing and evenly distributed traffic loads on optimal relay placement is provided in chapter 5.

1.5.2 Multicast and Broadcast

Multicast and broadcast are two other communication patterns residing at the network procedural level. Per multicast, information can flow within a "many-to-many" or "one-to-many" scenario: respectively, a subset of source nodes send data to a subset of destination nodes; and a single node sends a packet to multiple nodes in the network. In both scenarios the destination nodes may be multiple-hops away from the source node(s). Broadcast, on the other hand, requires that a message sent by a source node should reach *all* of the network nodes (please refer to Fig. 1.4).

In a sense, multicast could be thought of as the most general communication pattern in networks. Both unicast and broadcast can be viewed as special cases of multicast. However, due to the specific requirements of broadcast and unicast communication, the respective algorithms differ significantly from the multicast approaches. Typically multicast protocols rely on a collection of trees each spanning (possibly disjoint) subset of network nodes. Each tree is rooted at a source node and covers the multicast destination nodes. This approach was first proposed for wireless multihop networks multicast in [81], where authors built on earlier ides for multicast in wireline LANs developed in [82]. Notice that simple unicast protocols build a single-path or, in the case of multi-path routing, a collection of paths (potentially disjoint) between a single source and a single destination.

In contrast, as we see in chapter 4, the broadcast solution minimizing the number of re-transmitting nodes in the network is equivalent to finding a Minimum Connected Dominating Set (MCDS) in the wireless network graph. As discussed in chapter 4, minimizing the number of re-transmitting nodes has a number of benefits for network performance. Since broadcast is frequently invoked network primitive in many network layer protocols (for example, all unicast routing protocols in the on-demand and hybrid classes discussed above use a form of broadcast to build source-destination routing paths), it is important that a broadcast message is propagated within the network with low latency and utilizing minimum amount of resources (e.g. energy). Intuitively, the lower the number of nodes re-transmitting the broadcast message, the lower the latency and energy consumed in the network. Finding the MCDS in a general graph is an NP-hard problem ([83]). In unit disk graphs (UDG) utilized to model wireless networks, the hardness of the problem is preserved. Hence, efficient distributed approximation algorithms and heuristic solutions are typically sought to solve the broadcast problem. Chapter 4, discusses in detail state-of-the-art algorithms and a novel approach of broadcast transmissions prioritization in time, to achieve close to optimal performance in terms of minimizing the number of payload packet re-transmissions required to cover all network nodes.

CHAPTER 2

STRUCTURING NETWORKS AROUND THE DATA LEVEL

As noted above, communicating data is among the most energy expensive routines across different types of wireless networks. For instance, receiving and transmitting data in wireless sensor networks (WSN) consisting of Mica2 nodes running TinyDB applications constitutes about 59% of the total energy consumption [8]. In various applications running on similar WSN node platforms, only transmissions account for up to 50-65% of energy expenditure, depending on radio throughput [9]. The trend is similar for small WiFi connected devices, where radio operation may claim 70% of the total energy budget [10]. Reducing the amount of data transmitted and/or energy consumed per transmission could lead to longer network lifetimes and significantly impact network functions.

The focus of the next sections is on a particular communication scenarios in the context of wireless networks' energy efficiency. Suppose a group of source nodes have access to the same or highly correlated pieces of data that need to be sent to a common sink. This scenario may be inherently embedded at the data level, on one hand. For instance, a physical phenomenon changes slowly across space, and the nodes in a WSN have similar readings. This is true in a number of WSN deployments, where clusters of spatially proximate nodes sense and potentially transmit very highly correlated, almost identical values to the sink ([21],[84]). Also, due to the broadcast nature of wireless transmissions, a number of collocated nodes may receive the same data packet from a source node, while having the destination node as a common neighbor. On the other hand, the scenario may be proactively sought after by the network designer. I.e., nodes can be intentionally placed in clusters, so that all nodes within a cluster have access to "approximately common" data. Suppose there are $|G|$ nodes placed within a cluster $|G|$ and each node i needs to transmit a message $m_i, 1 < i < |G|$ at time t to sink s. Nodes within cluster G are said to have "approximately common data", if at any time t the difference (measured by some metric such as hamming distance, binary value, etc.) between any two packets is bounded by a constant:

$$\|m_i - m_j\| < b, \forall i, j, \text{ where } b \in \mathbb{R}^+ \tag{2.1}$$

For example, the operation of cooperative transmission networks (e.g. [85], [86]) and the related distributed transmit beamforming physical layer protocols [34] requires a specific case of approximately common data across a cluster of nodes that synchronizes their tranmissions. In the case of these protocols, b is 0 in (2.1). I.e. the nodes have access to common data.

Encoded sensing (ES) is a different novel scheme outlined and analyzed in [87] and [88] that substantially reduces the energy required for transmission of data in various types of wireless networks. ES operates by exploiting approximately common data embedded in the data level, or by structuring the network topology, so that approximately common data across multiple clusters of nodes is attained, and then exploited. ES reduces at least two-fold the number of radio signals required for conveying a message to the destination from each cluster of nodes. Furthermore, the received messages distort the original source messages within QoS constraints. The next section first provides a brief description of encoded sensing's basic idea in the context of a WSN, since ES is most intuitive in that setting.

2.1 Encoded Sensing for Energy Efficient Wireless Networks

Suppose a set N of sensor nodes is deployed within an event area $\mathcal{A}$, according to some spatial distribution Γ. The event source is continuous and governed by an apriori known spatial statistics. The source statistics model could be dictated by the phenomenon nature, or learned via training. Although measurements at different positions in $\mathcal{A}$ are frequently assumed i.i.d in the research literature, it is more realistic to assume that measurements are instead spatially-correlated and dependent on the distance between sensors ([89], [21]). The specifics of the spatial-correlation model are described in section 2.4.1, where we discuss further the application of ES to WSN.

Assume time is discretized in slots. The sensors sample $\mathcal{A}$, and a set of active nodes report measurements to the sink during each timeslot. Nodes are coarse-grained synchronized at a resolution of a timeslot. Suppose nodes transmit over multiple access AWGN channel at the same average power per bit, and that the sink B is within the transmission radius R of the nodes. The latter assumption is made for simplicity of presentation. ES can be readily extended to a multihop scenario.

Finally, imprecisions in sensors' electronics often occur in practice, introducing noise in each node's measurement. The statistical model for this instrumentation noise is provided in section 2.4.1.

WSN problem statement: under the above assumptions, one seeks a scheme that minimizes the energy spent in communicating measurements to the sink, while simultaneously guaranteeing a level of accuracy in estimating the value of the measured source. This minimum level of accuracy is prescribed to the system by QoS constraints. Sink's estimate accuracy is modeled via a standard Minimum Square Error (MSE) distortion metric fully specified in section 2.4.1.

2.1.1 Encoded Sensing Solution Intuition

Suppose the continuous source (e.g. a phenomenon or modality such as temperature) is quantized into a set of short intervals. The intervals are assigned to an index $\mathcal{I} = 1, 2, \ldots, M$. For simplicity assume that at a given timeslot, all sensor nodes in a group $G \subset N$ measure phenomenon values falling in the same interval $x \in \mathcal{I}$. We relax this assumption later. The goal of group G is to transmit measurement x to the sink, reliably and with minimum energy.

Per encoded sensing, the nodes in G are labeled with IDs: $0, 1, \ldots, |G| - 1$ as shown in fig. 2.1. Every interval $x \in \mathcal{I}$ is assigned to a distinct set of nodes $A_x \subset G$. Each of the nodes in $A_x \subset G$ assigned to x transmits a signal "1" to the sink. The nodes in G/A_x remain silent. Each transmitted signal "1" carries a "signature" of its source node $j \in A_x$. Receiving the signed signals, the sink can identify all j's in A_x. Assuming it knows the assignment of sets A_x to intervals $x, \forall x \in \mathcal{I}$, the sink can recover x. Since x is short, it is an accurate estimate of the value measured by the nodes in G.

In essence, the sink receives a "collaborative codeword" c_x of "1"s that encodes the interval x. Upon receiving a valid codeword c_x, the sink can decode it and obtain x. The number of available codewords depends on the cardinality of A_x and the size of the group G. For instance, given a group of size $|G| = 14$ and equal codeword lengths $|A_x| = |c_x| = 4$ one could generate $\binom{14}{4} = 1001$ different codewords; if $|G| = 17$, there are more than double, $\binom{17}{4} = 2380$, codewords. In

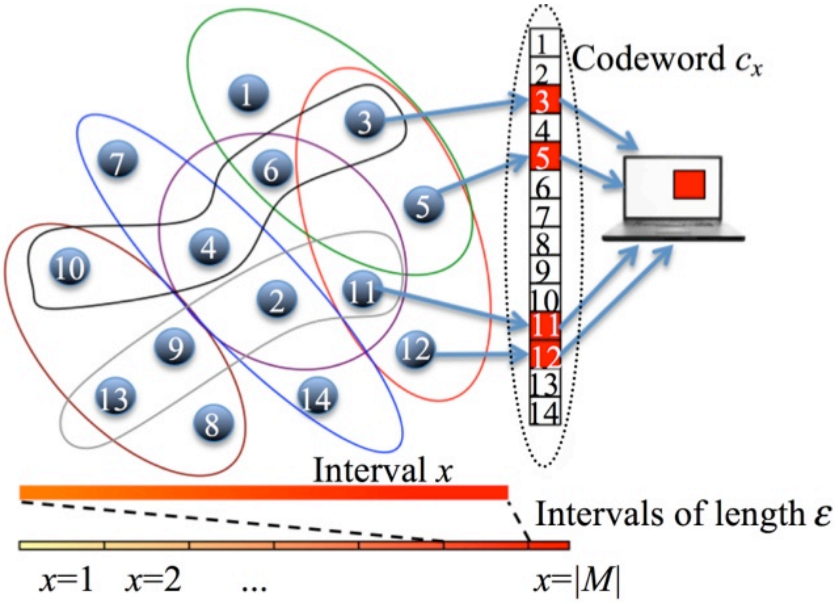


Figure 2.1: All nodes in the group G sense a value in the same small interval x (in solid red gradient). Only the 4 nodes (circled in red line) that are assigned to x transmit to the sink; the remaining 10 nodes are silent (white blocks). The sink identifies the 4 transmitting nodes (red blocks) and recovers the interval x.

the limit case, $|A_x| = |c_x| = 1$: given $|G| = M$ the measurement can be encoded over a single bit sent by a single source node j, given j's "signature". Notice that this limit case is somewhat analogous to utilizing M-ary orthogonal codes (e.g. PPM) to achieve reliable communication at the minimum possible energy per bit E_b as M increases ([90]). There, the total power is spread over a large time interval. As we will see, in the encoded sensing limit case, where $|G| = M$, the energy per bit is spread over nodes' specific orthogonal code "signatures".

Although most energy efficient, the limit case is not practical, as M could be on the order of tens of thousands and larger. However, one can observe a trade-off between the number of nodes in the network and network's energy efficiency. Given the number of nodes $|G|$ as a system constraint/parameter, the minimum codeword length $|A_x| = |c_x| = K$ is found, so that $\binom{|G|}{|K|} \geq M$, thus minimizing energy consumption for communication in group G. Also, a WSN of N nodes can be partitioned in multiple groups of nodes, so that encoded sensing employed locally in each group maximizes network-wide energy efficiency, while satisfying quality of service constraints.

In the above description, there are a few implicit assumptions regarding ES operation. In what follows we show how these assumptions can be met or relaxed; we study and quantify the benefits of employing ES; and we describe how ES may be generalized outside the scenario of WSNs.

- *Encoded sensing* ***collaborative*** *codes*:

• First, it is assumed that the sink and nodes "know" of a mapping where any message x is assigned to a distinct set of nodes $A_x \subset G$. Sections 2.1.2 and 2.2 provide and analyze decentralized algorithms for finding such mappings and constructing novel collaborative codes for energy efficient communication.

• In the case of WSN, it is also assumed that nodes in the same group G measure values falling in the same interval $x \in \mathcal{I}$. In practice, this may not always be the case due to measurement imprecisions, for instance. Section 2.2 (Theorems (2.2) - (2.4)) propose and analyze a novel Minimal Distance Combinatorial Encoding allowing nodes within a group to erroneously measure values falling in an interval x' differing from the true interval x. The nodes in the group still can transmit the correct collaborative codeword for interval x. The same ES codes can be used to encode general data messages instead of intervals x.

- *Encoded sensing signatures and* ***sparse*** *DSSS*:

• Next, it is assumed that the sink can identify each transmitting node's "signature" based on a single bit. Section 2.1.2 discusses a node identification technique at the sink based on standard direct sequence spread spectrum (DSSS)

PN sequences. A certain sparseness property of ES provides insight for a novel, significantly reduced, receiver circuitry (and cost) for DSSS devices based on compressive sensing sparse support recovery. Using the latter technique, the sink identifies nodes' signals with high probability (Theorem (2.5)).

Support recovery is sufficient for ES operation; ES does not require detection of the received symbol sign (i.e. detecting if the received bit is 0 or 1). However, the described sparse DSSS receiver design is potentially applicable in other, non-ES, DSSS deployment scenarios, too. Hence, for completion, we discuss a MMSE-based scheme for detecting the received symbol sign at the sparse DSSS receiver. The bounds for the conventional DSSS receiver design's symbol detection probability of error are derived and compared with the respective performance of the suggested sparse DSSS MMSE-based scheme.

- *Energy efficiency and distortion*:

- In section 2.4, ES is studied in a WSN application. ES achieves comparable energy gains to state-of-the-art cooperative distributed transmit beamforming schemes [34]. Unlike beamforming, ES does not reside on the physical layer and does not require fine synchronization across transmitting nodes, the latter often being impractical. Also, we show that ES is at least 2 times more energy efficient than non-cooperative (e.g. algorithms in [21] and other duty cycles schemes), where only one of the nodes in a group G transmits message x. The analytical rationale for the latter is given in section 2.5.

- The energy efficiency of the ES scheme is achieved while matching and guaranteeing optimal level of estimate distortion at the sink, within WSN's QoS constraints. Section 2.4.2 outlines a vector quantization scheme for determining the sites of representative groups of nodes with highly correlated measurements so that the resulting distortion is optimal. The scheme can be utilized orthogonally to ES to reduce distortion of measurement either in the context of topology control by placing nodes at the representative groups sites; or, for forming groups of highly correlated nodes, given network nodes have already been placed.

2.1.2 Encoded Sensing Stages

For clarity, consider a single group of nodes $G \subset N$. The nodes in G need to send message x to a single sink. Encoded sensing consists of four major stages: *encoding*, *assignment*, *transmission*, and *decoding*.

- At the *encoding stage*, each node in G locally encodes x into binary codeword c_x of length $|G|$, according to a common decentralized coding algorithm run at all nodes. We discuss such codes, satisfying different properties below. Let $c_x(j), 0 \leq j \leq |G| - 1$, denote the j-th most significant bit in c_x. Each node forms the set $A_x = \{j : c_x(j) = 1\}$.

- At the *assignment stage* each node j determines whether it is in the set A_x:

$$D_j = \begin{cases} 1, & \text{if } j \in A_x \\ 0, & \text{otherwise} \end{cases} \tag{2.2}$$

Node j assigns itself to x, iff $D_j = 1$.

- Next, at the *transmission stage*, for each node $j \in G$, if $D_j = 1$, j transmits a single bit "1_j" with j's signature, so that upon receiving "1_j" the sink can detect the sender node j. Otherwise, if $D_j = 0$, node j remains silent.

- Finally, at the *decoding stage*, the sink receives the set $\left\{1_j : j \in A_x\right\}$. The sink determines the identities of the transmitting nodes based on their signatures and recovers A_x. The codeword entry $c_x(j), 0 \leq j \leq |G| - 1$, is set to 1 if $j \in A_x$ and to 0 otherwise. The sink runs a decoding algorithm on c_x to obtain message x. The codeword c_x received at the sink has been collaboratively constructed by the nodes of group G. Such codewords are referred to as collaborative codewords.

Minimizing $|A_x|$ reduces the number of nodes that transmit signals, conveying message x to the sink, in turn minimizing communication energy. Notice that $|A_x| = w(c_x)$, where $w(c_x)$ is the weight of codeword c_x. The weight of a binary codeword as usual equals the number of 1's in it. The objective of a collaborative code would be to have a sufficiently large range C of codewords with length $|G|$, so that $C \geq M$. Simultaneously, $w(c_x)$ should be as small as possible for all x. Since each x must be encoded with a distinct codeword c_x, the subsets A_x of G have to be distinct. As an illustration of encoding sensing's mechanism, the next three subsections discuss a simple combinatorial code, along with respective encoding/decoding algorithms, that achieves optimal performance for sufficient number of nodes in a group G. However, this combinatorial code may not be practical in certain scenarios. Section 2.2, develops a more sophisticated code preserving the performance of the simple combinatorial code, while avoiding its limitations.

To reduce the number of nodes participating at the encoding and transmission stages and achieve optimal energy efficiency, the goal is to minimize $|A_x| = w(c_x)$, given M and $|G|$. Suppose M and $|G|$ are known locally at all nodes. The constraint here is that all M messages have to be assigned to M distinct codewords. Assume equiprobable messages at present, $|A_x| = w(c_x) = K, \forall x$. Then, notice that the number of possible distinct codewords is $\binom{|G|}{K}$; it is required $\binom{|G|}{K} \geq M$. For a given fixed group size $|G|$, one first finds

$$K^* = \min_{1 \leq K \leq |G|/2} K \text{ s.t. } \binom{|G|}{K} \geq M \tag{2.3}$$

and next constructs a code, where $w(c_x) = K^*$, $\forall x$. The range C of this code is $\binom{|G|}{K}$. For any given $|G|$ and M, K^* is easily computed numerically at each node. Picking K^* for the size of the subset of nodes A_x assigned to each message x guarantees that the number of nodes participating in the encoding of x is the minimum possible.

Each message x is indexed so that: $0 \leq x \leq M - 1$. Given message x, in the encoding step, each node runs **Algorithm** (2.1) locally. **Algorithm** (2.1) performs basic combination unranking. Hence this simple code is dubbed combinatorial encoded sensing (ES-C). For each distinct input pair (x, K^*) ES-C outputs a distinct set A_x of K^* integers taking values from 0 to at most $|G| - 1$. Stated differently, each distinct message x is mapped to a set A_x containing the IDs of the nodes in G responsible for x. A_x is the minimal possible such subset under the constraints of (2.3). At the assignment stage, each node $j \in G$ knows the set A_x assigned to the message x, and j knows its own ID. If j's ID is in A_x, j sets $D_j = 1$ in (2.2). Notice that without message passing between each other, the nodes in G have at this point locally and collectively encoded the message x to a codeword c_x, where $c_x(j), 0 \leq j \leq |G| - 1$, is set to 1 if $j \in A_x$ and to 0 otherwise.

Algorithm 2.1: COMBINATORIALASSIGNMENT

```
input: value of message $x$, $K^*$
output: assignment $A_x = \{n_{K^*}, n_{K^*-1}, \cdots, n_1\}$
$A_x \leftarrow \{\emptyset\}$
$m \leftarrow K^*$
// Each iteration, add node ID to the assignment $A_x$ of nodes to message $x$
while $m \geq 1$ do
    $n_m \leftarrow$ maximum integer $n_m$ such that $\binom{n_m}{m} \leq x$
    $x \leftarrow x - \binom{n_m}{m} \leq x$
    $m \leftarrow m - 1$
    $A_x \leftarrow A_x \cup n_m$
end while
```

2.1.3 Transmission and Optimality

At the transmission stage, the nodes in A_x can transmit their respective single-bit signals "1_j". The sink can only recover the codeword c_x if nodes' transmissions indicate nodes' identities as well. This information however is often implicit in many standard physical layer protocols employing variants of spread spectrum wave signatures embedded in each node's signal. For instance, the physical layer of IEEE 802.15.4 utilizes Direct Sequence Spread Spectrum (DSSS). Similarly practical and efficient DSSS architectures for low-powered WSNs have also been discussed in [91]. Long Term Evolution 3GPP networks are another example where each node is assigned a distinct, in this case, Zadoff-Chu sequence [92]. In general, some form of a signature identifying nodes' signals at the receiver is a common feature in many "off-the-shelf" multiuser systems ([93], [94]). In the above instances, each node's signal (bit) is multiplied by the node's unique sequence and transmitted. The received signal is correlated at the sink, via a bank of matched filters, with the set of available sequences' waveform signatures. The sink can obtain the identity of the transmitting nodes in this sequence acquisition stage. This is the only assumption ES makes regarding the physical layer. Section 2.3.1 expounds different possible DSSS designs.

Notice that given $|G| \geq M$ each message x can be encoded over a single bit, $w(c_x) = K^* = 1$, sent by a single source node j, where the bit is spread over j's DSSS spreading sequence. This limit case is similar to utilizing M-ary orthogonal codes (e.g. PPM) to achieve reliable communication over AWGN channels at the minimum possible energy per bit E_b, as M increases [90]. There, the total power is spread over a large time interval to achieve the Shannon limit of E_b/N_0 = -1.6dB. Here, DSSS spreading (e.g. PN, orthogonal, etc.) sequences have very low cross-correlation; ideally they are orthogonal to each other. In the ES-C limit case, where $|G| \geq M$ and $w(c_x) = 1$, the energy per the single bit transmitted over AWGN is spread over the spreading sequence of the single channel user j. Node j's sequence is orthogonal to the spreading sequences of the nodes in $G/\{j\}$ and ES-C operates at the optimum energy per bit level.

2.1.4 Decoding

After receiving and correlating nodes' transmissions exploiting spread spectrum physical layer properties, the sink identifies the nodes j in A_x each of which has transmitted only a single bit "1_j". Assuming the system operates under DSSS acquisition capacity, so that w.h.p. there are no errors in transmission, the sink can decode the codeword c_x. For completion, the acquisition capacity of a standard DSSS system is discussed in the Appendix A. The following theorem ensures that the IDs of the nodes in A_x available at the sink are sufficient to recover the index of the message x.

Theorem 2.1. *For every number $x \in \mathbb{N}$, $\exists$ a unique set $n_m, n_{m-1}, \ldots, n_1$, $n_i \in \mathbb{N}$ and $n_m > n_{m-1} > \cdots > n_1 \geq 0$, such that for any $m \in \mathbb{N}$, where $m \leq n_m$*

$$x = \binom{n_m}{m} + \binom{n_{m-1}}{m-1} + \binom{n_{m-2}}{m-2} + \cdots + \binom{n_1}{1} = \sum_{i=1}^{m} \binom{n_i}{i} \tag{2.4}$$

Proof. Please refer to [95]. □

Notice that Theorem (2.4) guarantees existence as well uniqueness. Namely, after ordering the set of IDs in A_x into the sequence $(n_m, n_{m-1}, , n_1)$ and having $m = K^*$ we obtain that

$$x = \sum_{i=1}^{m} \binom{n_i}{i} \tag{2.5}$$

The ES decoding process is simply a combination ranking of A_x.

2.1.5 Example of a Basic ES-C Run

Suppose a group of $|G| = 6$ nodes is required to send, among a set of $M = 16$ messages, the message $x = 0101_2 = 5_{10}$. It is easy to compute that $K^* = 3$ from (2.3). (By inspection, if $K^* = 2$, $\binom{6}{2} = 15 < M = 16$ and (2.3) is not satisfied.) Then, at the encoding step, all 6 nodes run **Algorithm** (2.1) with input pair $x = 5$ and $K^* = 3$. The output of **Algorithm** (2.1) is the set $A_5 = 4, 2, 0$. Nodes in G with IDs 4, 2 and 0 are assigned to $x = 5$. At the assignment stage, the binary decisions from (2.2) of the nodes in G are $D_4 = 1, D_2 = 1, D_0 = 1, D_5 = 0, D_3 = 0, D_1 = 0$. The transmitted collaborative codeword is $c_5 = 010101$. At the transmission stage each node in A_5 sends a single bit utilizing a physical layer allowing the nodes' identification at the receiver (e.g. DSSS as discussed in the previous section). After receiving the signals and identifying the transmitting nodes' IDs in A_x, the sink can decode the codeword c_5 using (2.5). Knowing $m = K^* = 3$ and ordering the IDs in A_x so that $(n_1, n_2, n_3) = (4, 2, 0)$ the sink obtains

$$\sum_{i=1}^{3} \binom{n_i}{i} = \binom{4}{3} + \binom{2}{2} + \binom{0}{1} = 5_{10} = 0101_2$$

This is the correct message and it has been transmitted utilizing only 3 signals.

ES-C is significantly more energy efficient than current communication schemes deployed in WSNs, as discussed in sections 2.4.3 and 2.5. However, ES-C implicitly requires that all nodes in group G have access to message x that needs to be sent. This need not be the case in the context of WSNs where each message x is uniquely mapped to one of M intervals spanning a continuous source. Nodes in the same group may measure different values falling in in-

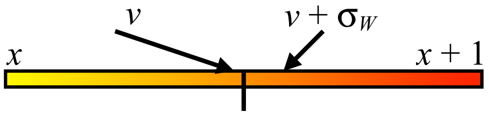


Figure 2.2: The value v of the phenomenon falls close to one of the endpoints of interval x. Due to sensing instrumentation imprecision σ_W, the value is incorrectly measured to fall in interval $x + 1$.

tervals x and x'. Independent on the source quantization (i.e. the length of the intervals), the probability of the latter event is not zero. Figure 2.2 illustrates the scenario leading to nodes in the same group reporting different measurements due to instrumentation noise σ_W. Theoretically we cannot preclude the event shown in fig. 2.2. The actual value v of the phenomenon may be arbitrarily close to either of interval x's endpoints, and even small instrumentation noise may lead some number k of nodes in G to measure values in the adjacent but wrong interval x'. The resulting ES-C collaborative codewords sent to the sink may be invalid or, worse, wrong causing large distortion in the sink's estimate even if a single node is in error. The instrumentation imprecision of sensors is modeled statistically in the technical literature as an additive Gaussian random variable $N(0, \sigma_W)$, independent at each node. We can set the length ϵ of each interval x so that the probability of a measurement being shifted to more than one intervals due to sensing imprecision is bounded and very low (i.e. by setting ϵ to a couple of standard deviations σ_W).

If a number k of nodes in a group G erroneously measure a value v' falling in interval x' and the rest $|G| - k$ nodes sense a value v falling in the correct interval x, could one find an encoding scheme with the property that almost surely the correct codeword is received at the sink, given $|x - x'| = 1$?

This question is answered in the positive by the minimum distance combinatorial encoding scheme presented in the next section. The scheme avoids the above shortcoming of ES-C, but preserves ES-C's energy efficiency.

2.2 Minimum Distance Combinatorial Encoding

Suppose there are two intervals x and x' within the phenomenon's range of values such that $|x-x'| = 1$. As above, x and x' are assigned to two distinct binary codewords. However, the assignment/encoding algorithm is different. It is required that if $|x - x'| = 1$, then the hamming distance between the codewords assigned to x and x' is bounded by a constant. There is a constant number $|A_x|$ active nodes out of $|G|$ nodes in a group at any time slot, thus one can obtain a new codeword from a given valid codeword flipping at least two bits: one of the bits has to flip to 0 and the other bit has to flip to 1 to preserve the number of active nodes. (E.g. if only one bit flips to 1, $|A_x|$ would increase by 1; and if only 1 bit flips to 0, $|A_x|$ would decrease by one.) More specifically, let $HD(c_x, c'_x)$ be the hamming distance between the codewords c_x and c'_x assigned to measurements in intervals x and x' respectively. The requirement is that

$$|x - x'| = 1 \Leftrightarrow HD(c_x, c'_x) = 2 \tag{2.6}$$

Notice that similarly to ES-C, in this setup $|c_x| = |c'_x| = |G|$ and $w(c_x) = w(c'_x) = |A_x|$.

The following section discusses the benefit of such minimum distance combinatorial encoding (MDCE) with the latter property; it also provides the algorithms to construct MDCE codebooks, and encode/decode MDCE words.

2.2.1 MDCE Probability of Error

Suppose the event in fig. 2.2 occurs with probability p_e independently at each node. Namely, the phenomenon's actual value v at a sensor's node location is within, but close to, the boundaries of interval x. Due to instrumentation noise, the sensor node at that location erroneously measures value in the interval x' such that $|x - x'| = 1$. The probability P_k that k nodes are in error is given by

$$P_k = \binom{|G|}{k} p_e^k (1 - p_e)^{|G|-k} \tag{2.7}$$

Suppose node j is one of the k nodes that erroneously determines the collaborative codeword to be transmitted is c'_x. Node j checks the bit at position j in c'_x to determine its decision. Notice that node $j's$ decision would be erroneous only if the bit at position j in c'_x is erroneous. Since $|x - x'| = 1$ it is guaranteed that $HD(c_x, c'_x) = 2$ from (2.6). Also, $|c'_x| = |G|$. Hence, the probability p_j of the event E_j that the bit at position j in c'_x is erroneous is given by

$$p_j = \frac{2}{|G|} \tag{2.8}$$

Given k nodes in error, the probability P_R that either of the k nodes reports erroneously (causing at least a single error in the codeword received at the sink) can be bounded as

$$P_R = \bigcup_{j=1}^{k} E_j \leq \sum_{j=1}^{k} p_j = \sum_{j=1}^{k} \frac{2}{|G|} = \frac{2k}{|G|} \tag{2.9}$$

where the inequality follows from the union bound. Since the probability that k nodes are in error is P_k from (2.7), overall, the probability P_E of error in the message is given by

$$P_E = P_k P_R \leq \frac{2k}{|G|} \cdot \binom{|G|}{k} p_e^k (1 - p_e)^{|G|-k} \tag{2.10}$$

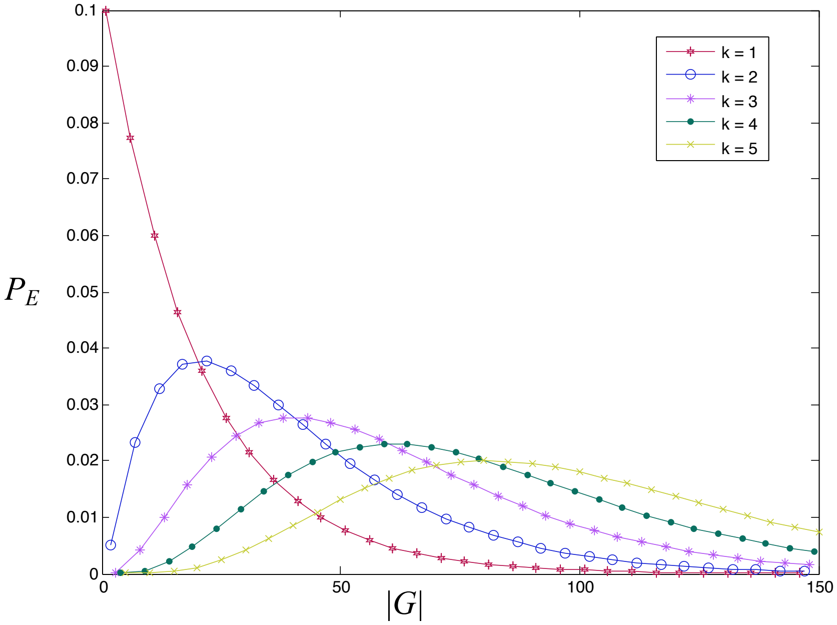


Figure 2.3: MDCE probability of encoding invalid or erroneous codeword given different number k of sensors with erroneous measurements, for $p_e = 0.05$.

Figure 2.3 shows the values of P_E for varying numbers of k and $|G|$. As the instrumentation noise decreases, the probability of inaccurate measurements decreases as well. Assuming p_e is small, k is a constant, and $|G|$ is large (2.10) becomes

$$P_E \leq \frac{2k}{|G|} \cdot \frac{e^{-|G|p_e} (|G|p_e)^k}{k!} \tag{2.11}$$

$P_E \rightarrow 0$ as $|G|$ increases, and the probability that a MDCE codeword transmitted by group G is invalid or wrong can be made arbitrarily low.

2.2.2 MDCE Construction

The MDCE construction developed below starts with the well-known Gray binary code with the property that the hamming distance between two consecutive codewords is one. Suppose a set of M messages must be encoded using MDCE and consider a Gray codebook with words of length $n = |G|$. Start with an empty MDCE codebook and let $x = 0$. For each consecutive word c in the Gray codebook, check if $m = w(c) = |A_x| = K^*$, so that (2.3) is satisfied; if true, add c to the MDCE codebook, setting $c_x = c$ and then incrementing x by one.

Theorem 2.2. *The resulting MDCE code satisfies* (2.6)*, and hence it is a minimum distance combinatorial encoding.*

Proof. First note that the standard reflexive binary Gray code of length n is given recursively as

$$Gray_n = 0Gray_{n-1}, 1reverse(Gray_{n-1})$$

where the *reverse()* operation simply reverses its input binary sequence. Let $MDCE_{m,n}$ be the subsequence of the $Gray_n$ code where $w(c) = m, \forall c \in Gray_n$. Then,

$$MDCE_{m,n} = 0MDCE_{m,n-1}, 1reverse(MDCE_{m-1,n-1})$$

Notice that $MDCE_{0,n} = \{000\ldots0\}$: a run of m 0s, denoted as 0^m. Also, $MDCE_{n,n} = 1^m$.

By induction, the first word in $MDCE_{m,n}$ is $0^{n-m}1^m$ and the last word in $MDCE_{m,n}$ is given by $10^{n-m}1^{m-1}$. For example, $MDCE_{1,2} = \{0MDCE_{1,1}, 1reverse(MDCE_{0,1})\} = \left\{0^{2-1}1^1, 10^{2-1}1^0\right\} = \{01, 10\}$, which is true.

Invariantly, to obtain

$$1reverse(MDCE_{m-1,n-1})$$

only 2 bits are flipped in the preceding

$$0MDCE_{m,n-1}$$

To see that, observe that by induction $0MDCE_{m,n-1} = 010^{n-m-1}1^{m-1}$ and $1MDCE_{m-1,n-1} = 110^{n-m-1}01^{m-2}, \forall m \geq 2$. Trivially, for $m = 1$ again only 2 bits are flipped to transition from $0MDCE_{1,n-1}$ to $1reverse(MDCE_{0,n-1})$.

Therefore every two consecutive words in the resulting code differ by the signs of two bits, and hence $|x - x'| = 1 \Leftrightarrow HD(c_x, c'_x) = 2$ is satisfied. Since the only selected codewords c from the Gray code are such that $m = w(c) = |A_x|$, the resulting code construction forms a minimum distance combinatorial encoding of the M messages. □

Notice that the MDCE codewords generated in this manner have weight equal to the energy efficient codewords of ES-C generated by **Algorithm** (2.1) for a given M and $|G|$. In fact, *the set* of MDCE codewords and that of ES-C are identical, for any given M and $|G|$. However, the mapping between codewords and messages is rather different.

Table (2.1) lists a subset of the MDCE and ES-C codewords for the case $|G| = 6, M = 20$, where $|A_x| = K^* = 3$ respectively. Notice that the hamming distance between two consecutive codewords is exactly equal to two.

The above MDCE generation procedure is not rather efficient, as it is based on the binary reflexive Gray codes, which are generated recursively. **Algorithm** (2.2) alleviates that problem and efficiently constructs MDCE codebooks for any given $|G|$ and K^*.

Theorem 2.3. ***Algorithm*** *(2.2) with input $K^* = |A_x| = m$ and $|G| = n$ generates a valid codebook $MDCE_{m,n}$.*

Proof. Consider the binary sequence of codewords in $MDCE_{m,n}$. Suppose each bit in each codeword is indexed, so that the first bit indicates whether node $n-1$ transmits, the second bit indicates whether node $n-2$ transmits, etc. so that the last bit indicates whether node 0 transmits. For instance, the binary codeword $1^5 1^4 1^3 0^2 0^1 0^0$ is equivalent to nodes 5, 4, and 3 transmitting; the binary $1^5 1^4 1^3 0^2 0^1 0^0$ can then be interpreted as the codeword 543. Then, one can convert

$$MDCE_{m,n} = 0MDCE_{m,n-1}, 1reverse(MDCE_{m-1,n-1})$$

to

$$MDCE_{m,n} \equiv MDCE^I_{m,n} = MDCE_{m,n-1}, \{n-1\} \cup reverse(MDCE_{m-1,n-1})$$

Let $(n_m, n_{m-1}, \ldots, n_2, n_1)$ be a codeword in $MDCE^I_{m,n}$. Note that the sequence of these codewords in $MDCE^I_{m,n}$ is sorted in the lexicographic order of

$$(n_m, -n_{m-1}, \ldots, (-1)^{m-1} n_1)$$

This follows directly by induction on the structure of the $MDCE^I_{m,n}$ code. It is straightforward to check that **Algorithm** (2.2) generates the codewords of $MDCE^I_{m,n}$ exactly in the lexicographical order of $(n_m, -n_{m-1}, \ldots, (-1)^{m-1} n_1)$. □

Table (2.1) illustrates the encoding of $MDCE^I_{3,6}$ generated by **Algorithm** (2.2) as an example. The basic combinatorial encoding from section 2.1.2 is also given in Table (2.1), for comparison. **Algorithm** (2.2) could be used to encode/decode measurements. However, that would require storing an assignment table similar to Table (2.1) at the sensor and sink nodes, and then looking up values of x and c_x respectively for encoding and decoding. The next theorem exploits the structure of the $MDCE^I_{m,n}$ code so that encoding and decoding are done much more efficiently, without requiring extra space for code tables.

Let the range of $MDCE^I_{m,n}$ be M, so that (2.3) is satisfied.

Algorithm 2.2: MinimumDistanceCombinatorialEncoding

```
input: $K^*$, $|G|$
output: list of assignments: $\{A_x = (n_{K^*}, n_{K^*-1}, \cdots, n_1)\} \forall x \in \mathcal{I}$
$m \leftarrow K^*$; $n_{m+1} \leftarrow |G|$; $L \leftarrow \{\emptyset\}$; $j \leftarrow m$
while $j \geq 1$ do
    $n_j \leftarrow j - 1$; $j \leftarrow j - 1$
end while
$j \leftarrow 1$
while $j \leq m$ do
    $L \leftarrow L \cup (n_m, n_{m-1}, \cdots, n_2, n_1)$
    $increment\_n_j \leftarrow false$
    $decrement\_n_j \leftarrow false$
    if $m$ is $odd$ then
        if $n_1 + 1 < n_2$ then
            continue // add assignment to $L$
        else
            $j \leftarrow 2$; $decrement\_n_j \leftarrow true$
        end if
    else
        if $n_1 > 0$ then
            $n_1 \leftarrow n_1 - 1$; continue // add assignment to $L$
        else
            $j \leftarrow 2$; $increment\_n_j \leftarrow true$
        end if
    end if
    while $true$ do
        if $decrement\_n_j$ then
            if $n_j \geq j$ then
                $n_j \leftarrow n_j - 1$; $n_{j-1} \leftarrow j - 2$
                break // add assignment to $L$
            else
                $j \leftarrow j + 1$; $increment\_n_j = true$
            end if
        end if
        if $increment\_n_j = true$ then
            if $n_j + 1 < n_{j+1}$ then
                $n_{j-1} \leftarrow n_j$; $n_j \leftarrow n_j + 1$
                break // add assignment to $L$
            else
                $j \leftarrow j + 1$
            end if
            if $j \leq m$ then
                $decrement\_n_j = true$
            else
                break // add assignment to $L$ and terminate
            end if
        end if
    end while
end while
```

Table 2.1: MDCE and Combinatorial Encoding Examples

x	**MDCE**	**MDCE**$^{I}_{3,6}$	**Combinatorial encoding**	**Combinatorial encoding index**
0	000111	2,1,0	000111	2,1,0
1	001101	3,2,0	001011	3,1,0
2	001110	3,2,1	001011	3,2,0
3	001011	3,1,0	001110	3,2,1
4	011001	4,3,0	010011	4,1,0
5	011010	4,3,1	010101	4,2,0
6	011100	4,3,2	010110	4,2,1
7	010101	4,2,0	011001	4,3,0
8	010110	4,2,1	011010	4,3,1
9	010011	4,1,0	011100	4,3,2
10	110001	5,4,0	100011	5,1,0
11	110010	5,4,1	100101	5,2,0
12	110100	5,4,2	100110	5,2,1
13	111000	5,4,3	101001	5,3,0
14	101001	5,3,0	101010	5,3,1
15	101010	5,3,1	101100	5,3,2
16	101100	5,3,2	110001	5,4,0
17	100101	5,2,0	110010	5,4,1
18	100110	5,2,1	110100	5,4,2
19	100011	5,1,0	111000	5,4,3

Theorem 2.4. *Any integer value x in the range of $MDCE^{I}_{m,n}$ can be uniquely represented as*

$$x = \sum_{j=1}^{m} (-1)^{m-j} \left(\binom{n_j}{j} - 1 \right)$$

where $(n_m, n_{m-1}, \ldots, n_2, n_1)$ is a codeword in $MDCE^{I}_{m,n}$.

Proof. Note that the tuple $(n_m, n_{m-1}, \ldots, n_2, n_1)$ is the x-th codeword generated by **Algorithm** (2.2), where

$$x = \sum_{j=1}^{m} (-1)^{m-j} \left(\binom{n_j}{j} - 1 \right) \tag{2.12}$$

To see that, consider the codewords $(n_j, n_{j-1}, \ldots, n_2, n_1)$ and $(n_{j-1}, n_{j-2}, \ldots, n_2, n_1)$ of the two codes $MDCE^I_{j,n}$ and $MDCE^I_{j-1,n'}$. Let the codeword $(n_j, n_{j-1}, \ldots, n_2, n_1)$ be the x-th codeword generated by **Algorithm** (2.2) and codeword $(n_{j-1}, n_{j-2}, \ldots, n_2, n_1)$ be the y-th codeword generated by **Algorithm** (2.2).

By induction,

$$x = \binom{n_j + 1}{j} - 1 - y, \forall j > 0$$

Then, starting with $j = 1$ and summing to $j = m$, using the above equation, one obtains

$$x = \sum_{j=1}^{m} (-1)^{m-j} \left(\binom{n_j}{j} - 1 \right) \tag{2.13}$$

□

Algorithm (2.3) and **Algorithm** (2.4), listed below, utilize directly Theorem (2.4) respectively to encode measurement x, in the range of code $MDCE^I_{m,n}$, and then decode x at the sink. Similarly to **Algorithm** (2.1), **Algorithm** (2.3) is run at each node of group G, upon measuring x. If node j's ID is in the output A_x then j transmits a 1. The sink receives the resulting 1's from all nodes in A_x (thus determining A_x). The sink then runs **Algorithm**(2.4) to recover x; **Algorithm** (2.4) computes the formula in (2.12).

Notice that both **Algorithms** (2.3) and (2.4) only work on a single codeword. The code table computed by **Algorithm** (2.2) is no longer required. In the case of $MDCE^I_{m,n}$, the runtime of **Algorithm** (2.4) is $O(m)$; here $m = |A_x| = K^*$. The worst case runtime input for **Algorithms** (2.3) is $|A_x| = K^* = 1$ and $x = M - 1$, yielding $O((M - 1)/|G|)$ performance. However, in practice the average runtime of **Algorithm** (2.3) is much lower depending on the distribution of x and the values of $|A_x|$ and $|G|$. The resulting scheme employing **Algorithms** (2.3) and

Algorithm 2.3: MDCEAssignment

```
input: value of message $x$, $K^*$, $|G|$
output: assignment $A_x = \{n_{K^*}, n_{K^*-1}, \cdots, n_1\}$
$A_x \leftarrow \{\emptyset\}$
$m \leftarrow K^*$
$n \leftarrow |G|$
// Each iteration, add node ID to the assignment $A_x$ of nodes to message $x$
while $m \geq 1$ do
    while $\binom{n}{m} > x$ do
        $n \leftarrow n - 1$
    end while
    $n_m \leftarrow n + 1$
    $A_x \leftarrow A_x \cup n_m$
    $x \leftarrow \binom{n+1}{m} - x - 1$
    $m \leftarrow m - 1$
end while
```

Algorithm 2.4: DecodeMDCE

```
input: assignment $A_x = \{n_{K^*}, n_{K^*-1}, \cdots, n_1\}$, $K^*$
output: value of message $x$
$m \leftarrow K^*$
if m is even then
    $x \leftarrow 0$
else
    $x \leftarrow -1$
end if
$c \leftarrow 1$
while $m \geq 1$ do
    $x \leftarrow x + c \cdot \binom{n^m}{m}$
    $c \leftarrow -c$
end while
```

(2.4) to construct and decode MDCE codewords is dubbed ES-MDCE.

2.3 Sparse DSSS receiver

This section explores an interesting sparseness property of the ES signal received at the sink. This allows for a novel DSSS design that significantly reduces the complexity and cost of the status quo DSSS receivers.

In regular DSSS systems, each PN sequence corresponds to a unique signature waveform $s_i(t)$ assigned to node i, $0 \leq t \leq T$ and $1 \leq i \leq N$, where T is the symbol duration and N is the total number of users in the system. At any given timeslot, the sink receives a signal $y(t)$ as a superposition of the transmitted signature waveforms. As shown in fig. 2.4A, the sink employs a set of N matched filters (MFs) for decorrelation of the received signals. Each MF_i correlates $y(t)$ with the signature waveform $s_i(t)$ of node i. Each MF_i comprises high complexity and high precision analogue circuitry driving the cost of a DSSS receiver proportional to the number of MFs/nodes in the system. Thus, along reducing circuit complexity, reducing the number N of MFs also reduces the actual cost of a DSSS receivers' deployment[1].

2.3.1 Sparse Support Recovery for DSSS

In a general setting, the task of reducing the number of MFs is hard, if at all feasible. We need information for all the N possible signature waveforms at the receiver in order to distinguish between the N different transmitted signals that can be received at any time. The crux here is that the ES communication scheme allows the number of signals transmitted and received during any timeslot to be much lower than N. Consider an ES system consisting of a single group of size $N = |G|$ with $|A_x| = K$ active nodes. We can encode $\binom{|G|}{K}$ different messages; for instance if $K = 3$ and $|G| = 40$ we can generate $\binom{|G|}{K} = 9880$ different codewords. At the receiver, each codeword contains exactly K non-zero bits and can

[1] The interested reader may find a more detailed description of DSSS systems and their properties in [93], [96], and [97] among others. A briefer but instructional discussion regarding traditional DSSS design and the important signal acquisition mechanism of DSSS-based system is given in the Appendix A of this work, as well.

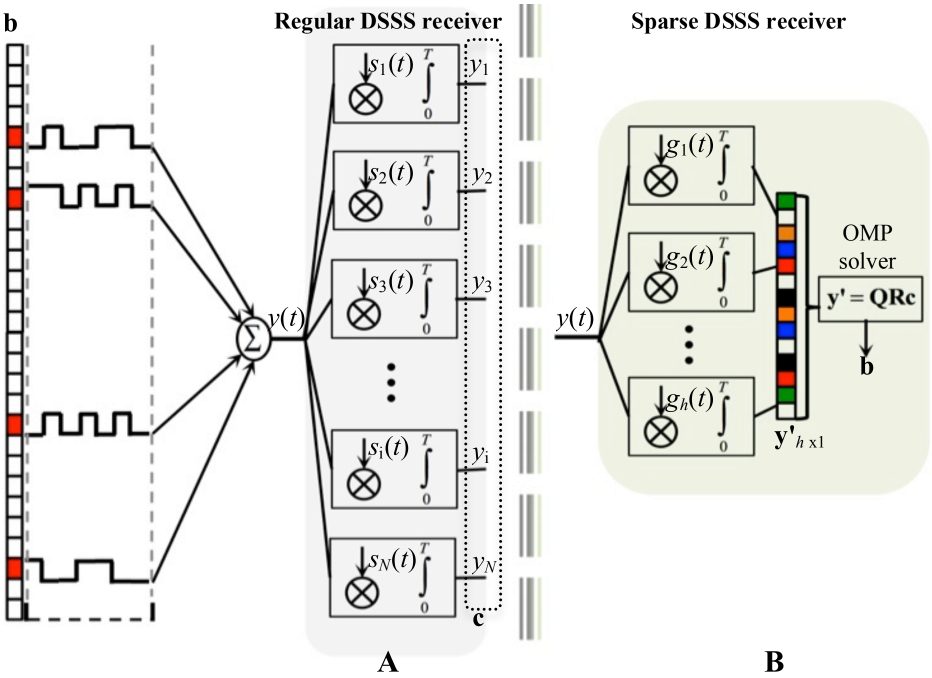


Figure 2.4: Receiving signal $y(t)$ with K-sparse support vector b. A: regular DSSS receiver consisting of N matched filters; B: sparse DSSS receiver consisting of $Klog(N/K)$ matched filters.

be represented as a binary K-sparse vector $\mathbf{b}_{Nx1}$. The received signal is

$$y(t) = \sum_{i=1}^{N} b_i s_i(t)$$

containing the signature waveforms of the transmitting nodes. The task of the MFs at the receiver is only to recover the *support* of **b** and thus find the identities of the transmitting nodes. As discussed above, this is sufficient for the operation of ES, which does not require symbol detection. Let l denote the length of the PN sequences. I.e. l equals the number of chips per PN sequence.

Theorem 2.5. *Given that ES is utilized, the support of* **b** *is recovered correctly w.h.p. at the DSSS receiver utilizing on the order of $h = Klog(N)$ matched filters, instead of N matched filters, iff $l > 2Klog(N)$.*

Proof. To show this, suppose each MF now correlates the received signal $y(t)$

with a signature waveform $g_j(t)$ (fig. 2.4 B), instead of utilizing the regular signature waveform $s_i(t)$. The signature $g_j(t)$ is obtained as follows:

$$g_j(t) = \sum_{i=1}^{N} q_{ji} s_i(t), \text{ where } 1 \leq j \leq h \tag{2.14}$$

here q_{ji} are the random ±1 entries of a Rademacher matrix $(\mathbf{Q})_{hxN}$. Now, there are h signature waveforms at h MFs. Each signature waveform $g_j(t)$ is a random linear combination of the original N signature waveforms. After receiving $y(t)$, the output of the j-th MF is given by the crosscorrelation between $y(t)$ and $g_j(t)$, namely the inner product $y'_j = \left\langle g_j(t), y(t) \right\rangle$.

The output of all h MFs, represented in vector form, is $\mathbf{y}' = \mathbf{QRb}$, where $(\mathbf{R})_{NxN}$ is the cross-correlation matrix of the original signature waveforms $s_i(t)$.

Due to the *universality property* and *restricted isometry property (RIP)* of $\mathbf{Q}$, the matrix $\mathbf{\Theta} = \mathbf{QR}$ also has the RIP w.h.p., if $h \geq \alpha K log(N/K)$, where α is a small constant ([98], [26]). Then the support of $\mathbf{b}$ can be recovered using a standard Orthogonal Matching Pursuit (OMP) (e.g. [99]). Applying the argument in [99], it is required that $l > 2K log(N)$, and this is sufficient for OMP to recover the support of $\mathbf{b}$ with high probability. □

Suppose the PN sequence has length $l = 32$ chips (similarly to PN sequences in IEEE 802.14.5). This implies that, for instance, with $K = 3$ and $N = |G| = 40$, one requires 16 MFs to identify the 3 transmitting nodes w.h.p., in comparison to 40 MFs in a regular DSSS receiver. This is a 60% reduction in circuitry! Notice that we have $l = 32 > 2 \cdot 3 log(40) = 31.93$ and satisfy the sufficient condition of Theorem (2.5). Also, note that a compressive sensing argument similar to the above can readily be devised in order to reduce significantly the *length of the* PN sequences generated in a DSSS system instead of reducing the complexity of the receiver. This could potentially increase the energy efficiency of ES further. We do not pursue this avenue here, but it could be of interest for future research. Furthermore, the latter observation may have implications for designing general DSSS-based systems with variable length PN sequences. The more the utilization of a system, the longer the PN sequences employed; once the utilization is lower the length of the PN sequences can be reduced significantly using an argument similar to the one in Theorem (2.5). Such design would benefit significantly the energy efficiency of devices deployed in rural areas, where CDMA-based cell towers service on average a small number of users.

2.3.2 Symbol Detection

The preceding subsection discussed how exploiting the sparseness of ES collaborative codewords, one can recover nodes' identities w.h.p at the sink, efficiently utilizing fewer MFs, which is sufficient for ES operation. However, other potential applications of the sparse DSSS receiver may utilize symbol detection (i.e. determining whether the bit transmitted by a given node *j* is 0 or 1).

2.3.2.1 Sparse DSSS Symbol Estimator

The DSSS receiver detector can be constructed using different approaches. The scenario here is very similar to multi-user detection (MUD). It has been shown that the optimal maximum likelihood detector for MUD is intractable in practice [100]. Therefore various lower complexity approximate detectors have been proposed (e.g. refer to section 13.4.4 of [97]). The MMSE estimator is in the category of efficient low complexity, linear MUD detectors. Therefore, the following section derives the *optimal* MMSE estimator for the sparse DSSS receiver described in the previous section.

The value of the bit that node j sends have so far been assumed to be 1_j, given $D_j = 1$ in (2.2). Instead, suppose now node j sends either a 0 or 1 (equivalently -1 or 1) when $D_j = 1$. The sink needs to detect the correct sign of the transmitted bit. That is, K nodes transmit in group G; next the identities of the transmitting nodes are recovered correctly, per Theorem (2.5); and finally, it is required that the receiver determine the sign of the bit transmitted by each of the K nodes. Once the support vector $\mathbf{b_S}$ of $\mathbf{b}$ is determined, errors can still occur while detecting each of the K bits in $\mathbf{b}$ due to multi-access interference (MAI).

Assume that the sparse support of the received signal has already been recovered correctly, as described in the previous subsection. Then, also notice that only the signs (-1 or 1) of the bits in the *support* of $\mathbf{b}$ need to be estimated; the rest of b_j's are 0.

As above, let $\mathbf{R}$ be the normalized cross-correlation matrix of the original $N = |G|$ signature waveforms:

$$\mathbf{R} = \left[\left\langle s_i(t), s_j(t)\right\rangle\right]_{i,j=1}^{N}$$

Suppose the channel gains $p_i, i = 1, 2, \ldots, N$, are known at the receiver and

$$p_i = a_i \sqrt{P_i}$$

where a_i is the channel amplitude also known at the receiver. Let $\mathbf{P}$ be a diagonal matrix with entries $p_1, p_2, \ldots, p_N$. The output of the sparse DSSS receiver in vector form is then given by

$$\mathbf{y}' = \mathbf{QPb} + \mathbf{z} \tag{2.15}$$

Here, $\mathbf{z}$ captures the effect of MAI and is a Gaussian r.v. with zero mean and covariance $\sigma_z^2\mathbf{QRQ}^T$. As per the typical MUD MMSE ([101]), the MMSE detector at the sparse DSSS receiver determines the sign of each bit by applying a linear transform $\mathbf{D}$ to the output of the h MFs. I.e.

$$\hat{b}_j = sgn\left(\left(\mathbf{Dy}'_j\right)\right) \tag{2.16}$$

so that the mean square error between $\mathbf{b}$ and $\hat{\mathbf{b}} = \left(\hat{b}_j\right)_N$ is minimized.

The challenge here is to find $\mathbf{D}^*$ for the sparse DSSS receiver so that

$$\mathbf{D}^* = \arg\min_{\mathbf{D}} E\left[(\mathbf{b} - \mathbf{Dy}')^2\right]$$

First, note that

$$E\left[(\mathbf{b} - \mathbf{Dy}')^2\right] = tr\left[E\left[(\mathbf{b} - \mathbf{Dy}')(\mathbf{b} - \mathbf{Dy}')^T\right]\right]$$

Carrying the multiplication, substituting $\mathbf{y'}$ from (2.15) and the covariance expression of $\mathbf{z}$, one obtains

$$E\left[(\mathbf{b} - \mathbf{Dy}')^2\right] = tr\left(\mathbf{I} + \mathbf{D}\left(\mathbf{QP}^2\mathbf{Q}^T + \sigma_z^2\mathbf{QR}^{-1}\mathbf{Q}^T\right)\mathbf{D}^T - \mathbf{PQ}^T\mathbf{D}^T - \mathbf{DQP}\right)$$

Suppose

$$\mathbf{D}^* = \mathbf{PQ}^T\left(\mathbf{QP}^2\mathbf{Q}^T + \sigma_Z^2\mathbf{QR}^{-1}\mathbf{Q}^T\right)^{-1} \tag{2.17}$$

It can be shown by straightforward arithmetic that

$$E\left[(\mathbf{b} - \mathbf{Dy}')^2\right] = tr\left(\mathbf{I} - \mathbf{D}^*\mathbf{B}(\mathbf{D}^*)^T + (\mathbf{D} - \mathbf{D}^*)\mathbf{B}(\mathbf{D} - \mathbf{D}^*)^T\right)$$

where $\mathbf{B} = \left(\mathbf{QP}^2\mathbf{Q}^T + \sigma_Z^2\mathbf{QR}^{-1}\mathbf{Q}^T\right)$. Substituting $\mathbf{D}^*$ from (2.17) in the second term above yields

$$E\left[(\mathbf{b} - \mathbf{Dy}')^2\right] = tr\left(\mathbf{I} - \mathbf{PQ}^T\mathbf{B}^{-1}\mathbf{QP} + (\mathbf{D} - \mathbf{D}^*)\mathbf{B}(\mathbf{D} - \mathbf{D}^*)^T\right) \tag{2.18}$$

Notice that $\mathbf{QP}^2\mathbf{Q}^T$ is a positive definite matrix. Also, the cross correlation matrix $\mathbf{R}$ is positive definite implying $\sigma_Z^2\mathbf{QR}^{-1}\mathbf{Q}^T$ is positive definite. Therefore, $\mathbf{B}$ is positive definite. Then, $tr\left(\mathbf{PQ}^T\mathbf{B}^{-1}\mathbf{QP}\right) > 0$. From (2.18) it follows that

$$D^* = \mathbf{PQ}^T\left(\mathbf{QP}^2\mathbf{Q}^T + \sigma_Z^2\mathbf{QR}^{-1}\mathbf{Q}^T\right)^{-1} = \arg\min_{\mathbf{D}} = E\left[(\mathbf{b} - \mathbf{Dy}')^2\right]$$

Thus, the MMSE detector's linear transformation is given by $\mathbf{D}^*$, and the receiver can readily estimate the signs of the transmitted symbols in the support of $\mathbf{b}$ using D^* in (2.16).

2.3.2.2 Conventional MUD Symbol Estimator

Let $Pr_j(error)$ be the probability of error in detecting node j's bit sign at the receiver. For completion, the bounds on $Pr_j(error)$ of the *conventional* multiuser detection receiver are derived next. The conditions that need to be met so that $Pr_j(error) \rightarrow 0$ are also obtained. Finally, $Pr_j(error)$ of the conventional symbol estimator is compared with the respective performance of the sparse DSSS receiver developed in 2.3.2.1.

The MMSE estimator is utilized in this case as well for MUD. Carrying over notation from the previous section, this time

$$\mathbf{y}' = \mathbf{RPb} + \mathbf{z} \tag{2.19}$$

Here, $\mathbf{z}$ is a vector representing only white Gaussian noise variables with power σ_Z^2, and now $\mathbf{y'}$ has N entries. Similarly to the above the MMSE detector determines

$$\hat{b}_j = sgn\left(\left(\mathbf{Dy}'_j\right)\right) \tag{2.20}$$

so that the mean square error between $\mathbf{b}$ and $\hat{\mathbf{b}} = \left(\hat{b}_j\right)_N$ is minimized. It is well-known (e.g. [100]) that the optimal linear transformation $\mathbf{D}^*$ here is given by

$$\mathbf{D}^* = \left(R + \sigma_Z^2 P^{-2}\right)^{-1} \tag{2.21}$$

From Proposition 3.1 in [101], asymptotically, as P_j/σ_j grows, the probability of error in detecting node j's sent bit is given by

$$Pr_j(error) = Q\left(\frac{P_j}{\sigma_Z \sqrt{\left(\mathbf{R}^{-1}\right)_{j,j}}}\right) \tag{2.22}$$

The impact of the PN sequence ensemble choice on the system performance is not immediately obvious from the term $\left(\mathbf{R}^{-1}\right)_{j,j}$ as given in (2.22). To elucidate this impact, it would be instructive to derive closed form bounds on this expression, explicitly as a function of the maximum cross-correlation between PN sequence signatures used in the system.

Lemma 2.1. *Let N be the number of users in the system and χ_{max} be the highest cross-correlation between any two PN sequences in the system. Then*

$$\left[1 + (N-1)\chi_{max}\right]^{-1} \leq \left(\mathbf{R}^{-1}\right)_{j,j} \leq \left[1 - (N-1)\chi_{max}\right]^{-1} \tag{2.23}$$

Proof. As observed earlier, the cross-correlation matrix $\mathbf{R}$ is positive definite and its eigenvalues can be ordered as $0 < \lambda_1 \leq \lambda_2, \ldots, \leq \lambda_N$. Let $\triangle \geq \lambda_N$ and $\triangledown \leq \lambda_N$. Then, from [102],

$$\frac{1}{\triangle} \leq \left(\mathbf{R}^{-1}\right)_{j,j} \leq \frac{1}{\triangledown} \tag{2.24}$$

Let $\mathbf{R} = \mathbf{E} + \mathbf{F}$, where $\mathbf{E}$ is diagonal matrix with entries $e_1, e_2 \ldots, e_m$ and $\mathbf{F}$'s diagonal entries are 0's. From *Gershgorin's Circle Theorem* (GCT) (e.g. [103]), if $\lambda(\mathbf{R})$ is the set of all eigenvalues of $\mathbf{R}$, then

$$\lambda(\mathbf{R}) \subseteq \bigcup_{i=1}^{N} D_i$$

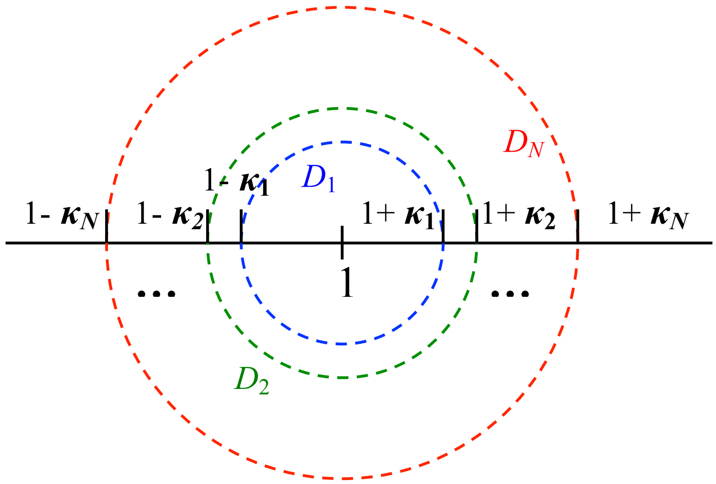


Figure 2.5: Possible positions of cross correlation matrix's eigenvalues on the real line. Here, interval $[1 - {}_N, 1 + {}_N]$ is described by the circle D_N.

where

$$D_i = \left\{ z \in \mathbb{C} : |d_i - z| \leq \sum_{j=1}^{N} |\,(\mathbf{F})_{i,j}\,| = \kappa_i \right\}$$

That is, all eigenvalues of $\mathbf{R}$, lie within the union of closed disks D_i, in the complex plane. Each disk D_i is centered at the corresponding diagonal entry of $\mathbf{R}$ and has radius equal to the sum of the non-diagonal entries at row/column i of $\mathbf{R}$. Here, all eigenvalues of $\mathbf{R}$ are real and D_i's are intervals on the real line centered around $\left(\mathbf{R}^{-1}\right)_{i,i} = 1$. Figure 2.5 illustrates the possible positions of $\mathbf{R}$'s eigenvalues. $\lambda(\mathbf{R})$ lie within the interval $D_N = [1 - \kappa_N, 1 + \kappa_N]$, where $\kappa_1 < \kappa_2 <, \ldots, < \kappa_N$. Then,

$$\kappa_N = \arg\max_{1 \leq i \leq N} \left\{ \sum_{j=1}^{N} |\,(\mathbf{F})_{i,j}\,| \right\} \leq (N-1)\,\chi_{max}$$

Suppose $\kappa_N = (N-1)\,\chi_{max}$. From GCT

$$\min\{\lambda(\mathbf{R})\} \geq 1 - \kappa_N = \triangledown > 0 \textbf{ and } \{\lambda(\mathbf{R})\} \leq 1 + \kappa_N = \triangle$$

Therefore,

$$\triangledown = 1 - (N-1)\chi_{max} \textbf{ and } \triangle = 1 + (N-1)\chi_{max} \tag{2.25}$$

Then, combining with (2.24) one obtains

$$[1 + (N-1)\chi_{max}]^{-1} \leq \left(\mathbf{R}^{-1}\right)_{j,j} \leq [1 - (N-1)\chi_{max}]^{-1} \tag{2.26}$$

□

Observe that as the highest cross-correlation between two PN sequences goes to 0 (as it should ideally), the entry $\left(\mathbf{R}^{-1}\right)_{j,j} \to 1$, from Lemma (2.1).

Theorem 2.6. *As P_j/σ_j increases, the asymptotic symbol detection probability, $Pr_j(error)$, of the conventional MMSE MUD detector is upper bounded by*

$$U = \frac{1}{12} exp\left\{\frac{P_j^2\,[(N-1)\chi_{max} - 1]}{2\sigma_Z^2}\right\} + \frac{1}{4} exp\left\{\frac{2P_j^2\,[(N-1)\chi_{max} - 1]}{3\sigma_Z^2}\right\}$$

and lower bounded by

$$L = \frac{1}{12} exp\left\{-\frac{P_j^2\,[1 + (N-1)\chi_{max}]}{2\sigma_Z^2}\right\} + \frac{1}{4} exp\left\{-\frac{2P_j^2\,[1 + (N-1)\chi_{max}]}{3\sigma_Z^2}\right\}$$

Proof. Using

$$Q(x) \approx \frac{1}{12} e^{\frac{-x^2}{2}} + \frac{1}{4} e^{\frac{-2x^2}{3}} \text{ (ref. [104])}$$

and substituting the result from Lemma (2.1) yields the bounds. □

Figure 2.6 plots numerically the symbol estimation probability of error for the conventional DSSS MUD detector based on MMSE (along with its upper bound) and the sparse DSSS receiver based on the MMSE with the optimal linear transform $\mathbf{D}^*$ derived in (2.17); $\sigma_Z = 0.2, P_j = 1, \chi_{max} = 0.02, N = 40$. The number of MFs varies in the case of the sparse DSSS receiver. Notice, that as the number of MFs increases the symbol estimation probability of error decreases for the sparse DSSS receiver and becomes comparable to that of the conventional DSSS receiver. Note that this is conditional on the support of **b** being correctly

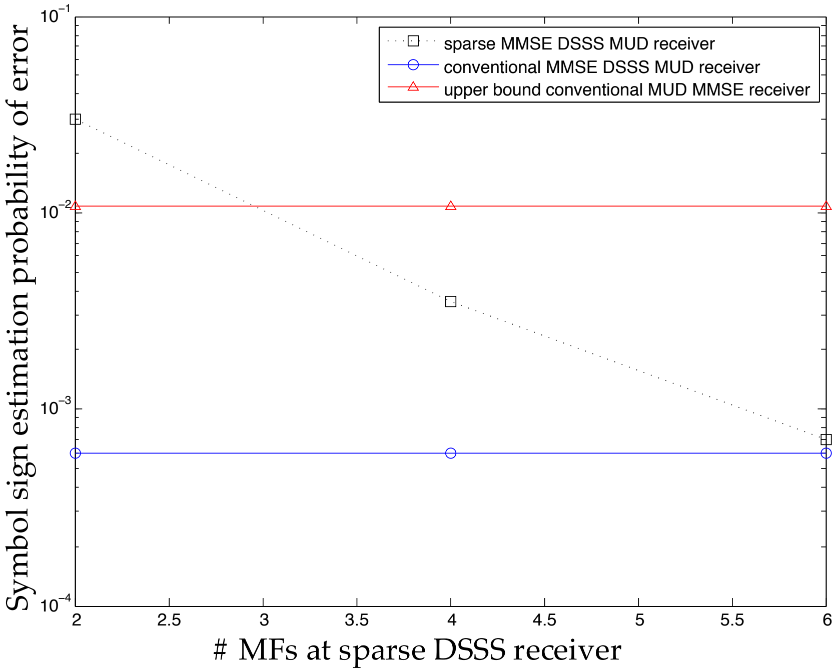


Figure 2.6: Symbol sign estimation probability of error for varying number of MFs at the sparse DSSS receiver's. The number $N = 40$ of MFs for the conventional receiver numerical performance and theoretical upper bound is a constant. Notice again that ES does not require symbol sign estimation and requires only support recovery as described in section 2.3.1.

computed. For that, one needs to deploy greater number of MFs than shown on the figure here. To emphasize again, the ES scheme, does not require the symbol sign estimation and support recovery, as given in section 2.3.1 suffices.

2.4 A Wireless Sensor Network Application

Next, encoded sensing's benefits are demonstrated in a practical WSN application. All N sensor nodes utilize ES to communicate their measurements with a sink.

2.4.1 Phenomenon and Sensing Models

Assume a continuous source **S** giving rise to a space-time random field $s(l, x, y)$. In the l-th timeslot the random field is defined as $\left\{S_j[l] = s(l, x_j, y_j) : (x_j, y_j) \in \mathcal{A}\right\}$ at spatial position (x_j, y_j), according to a spatial-correlation model. The instances $S_j[l]$ are modeled as joint Gaussian random variables (JGRV). Considering a single discrete-time interval sample, the time index l can be dropped. The JGRV is characterized by:

$$E\left[S_j\right] = 0, Var[S_j] = \sigma_S^2, \rho_{i,j} = E\left[S_i S_j\right]\sigma_S^2 \tag{2.27}$$

where $\rho_{i,j}$ are the correlation coefficients of the JGRV [2].

Often physical phenomena's values at different points in space are dependent on one another and are correlated via some function of the distance between them. Formally, this function is represented by a parameterized spatial covariance model, which reflects the nature of the specific phenomenon and determines the correlation coefficients. The covariance model selected here is the *Power Exponential* since different physical phenomena monitored by sensor networks could be approximated this way (e.g. [21], [105], [14]):

$$\rho_{i,j} = \mathcal{K}_\theta\left(\|\mathbf{i} - \mathbf{j}\|\right) = e^{\left(-\frac{\|\mathbf{i}-\mathbf{j}\|}{\theta_1}\right)^{\theta_2}} \tag{2.28}$$

where $\|\cdot\|$ denotes the Euclidean distance between sensors i and j at (x_i, y_i) and (x_j, y_j).

A sensor node j with coordinates (x_j, y_j) does not have direct access to the phenomenon value S_j. Rather, it obtains a distorted measurement X_j of S_j due

[2] We assume memoryless source and do not account for temporal correlations. However, ES is readily adjustable to temporal correlations: if some readings are more likely than others **Algorithms** (2.3) and (2.4) can be easily extended to allow for a round-robin assignment of nodes to message x in order to improve the energy balance of a group.

to inherent sensing imprecision/noise W_j. W_j's are assumed to be i.i.d. Gaussian random variables, such that $E\left[W_j\right] = 0$ and $Var\left[S_j\right] = \sigma_W^2$.

Ultimately, j measures $X_j = S_j + W_j$.

2.4.2 Encoding Sensing Setup

To utilize ES one needs to ensure that the network is partitioned in groups of $|G|$ nodes such that the nodes in each group have access to the same message x. To this end, there are two conditions we need to meet simultaneously.

- First, each group occupies a site where the actual values of the source at the different points by the sensors fall in the same interval $x \in \mathcal{I}$ w.h.p..
- Second, noise due to electronic imprecision should not offset node j's measurement in an interval x' different from x, for all $j \in G$, where $|x - x'| > 1$.

Source quantization: To meet the *second* condition, we quantize the interval $\mathcal{I}$ so that the probability is low that the Gaussian measurement imprecision W_j offsets the measurement of any $j \in G$ outside interval x', where $|x - x'| \leq 1$. Therefore, the size of each interval is set to $\epsilon = 2\beta\sigma_W, \beta \geq 2$.

Vector quantization: To meet the *first* condition, the network is partitioned in disjoint groups of highly correlated nodes. This can be achieved by utilizing the distributed vector quantization algorithm underlying the CC-MAC protocol presented in [21]. Given the nodes' spatial statistical distribution Γ and the spatial correlation model as an input, the algorithm selects k representative nodes out of the N nodes in the WSN. Per [21] , the selected k nodes are chosen so that the spatial correlation between their measurements is reduced. Each of the selected k nodes transmits (non-cooperatively) a measurement to the sink every timeslot; the rest of the N nodes' measurements can be eliminated. This describes the basic operation of a *non-cooperative transmission with decorrelation* (NCD) scheme. The k representative nodes are selected, so that distortion in the sink's estimate of the source $\mathbf{S}$ is *minimized to a QoS threshold*.

One can exploit a related but different property of the algorithm in [21] to satisfy the ES requirement that a group G of nodes all sense values in the same interval x. All nodes within distance $r_{corr} < r$ of each representative node, have highly spatially-correlated measurements and report almost identical values. r_{corr} is an output of the representative node vector quantization algorithm used in [21]. Consider a disk, O_i, with radius r_{corr}, centered at representative node i.

Definition 2.1. A group of sensor nodes, G_i, situated in disk O_i is called a *Representative Group*.

To summarize, nodes within each representative group G_i utilize ES-MCDE to transmit their measurements to the sink. Encoding sensing operates in each G_i, as described in sections 2.1.2 and 2.2: each short interval x is assigned to a distinct subset of nodes A_x in G_i according to **Algorithm** (2.3) . The length of the intervals is constant and set according to the **source quantization** discussed above. Upon measuring a value in interval x, each node $j \in G_i$ checks if it is in A_x. If so, j transmits a signal using DSSS. The sink receives the collaborative codeword c_x comprising the transmission signals of all nodes in A_x; finally, the sink recovers the interval x using **Algorithm** (2.4). In Appendix B, it is shown analytically that as the sensing imprecision due to W_j decreases, the distortion $D_{ES}(k)$ in the sink's estimate when ES is utilized converges to the distortion $D_{NCD}(k)$ that can be obtained within QoS constraints, via the optimal

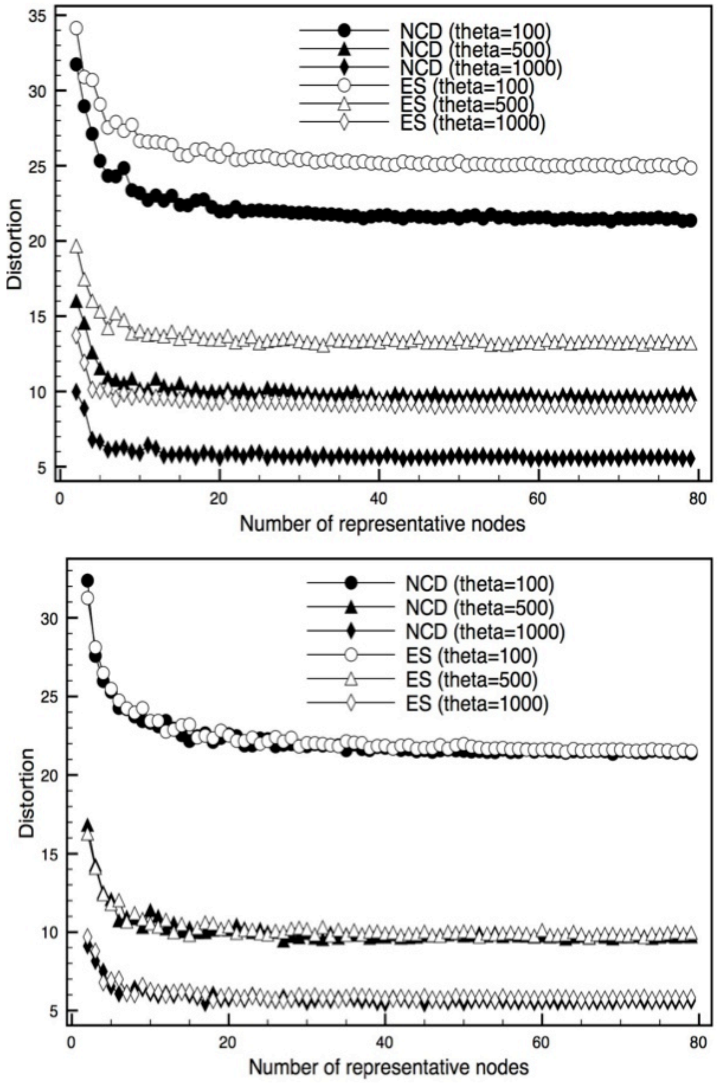


Figure 2.7: Distortion of measurement for ES and NCD; $\sigma_W = 0.25, 0.05$ respectively **top** and **bottom**; $\theta_S = 5$ throughout. ES and NCD converge for lower values of σ_W.

non-cooperative transmission of k representative nodes [21]. Figure 2.7 illustrates the numerical evaluation of $D_{ES}(k)$ and $D_{NCD}(k)$ derived in Appendix B. The behavior of the two distortion functions is evaluated for varying number of representative nodes/groups, given the covariance model (2.28) and its parameters: $\theta_1 \in \{100, 500, 1000\}$ and $\theta_2 = 1$.

2.4.3 Perfromance Evaluation

2.4.3.1 Simulation System

The simulation environment and schemes' implementations are programmed in JAVA and utilize the BLOG Inference Engine available online [106]. In each simulation run, N nodes are placed within a $100[m] \times 100[m]$ square area. S_j's are modeled as spatially correlated JGRVs with covariance model $\mathcal{K}_\theta$ (2.28).

- The phenomenon's values are assumed to be in the range $[-4\sigma_S, 4\sigma_S]$; the standard deviation is $\sigma_S = 625$.

- Notice that if nodes' sensors measure temperature with precision 0.1 degree, the range in the setup above would allow for temperature measurements between -250 and 250 degrees for all schemes. A sample output of the system is shown in fig. 2.8. The BLOG Inference Engine is used to generate the set of S_j's. The parameters of $\mathcal{K}_\theta$ are set to $\theta_1 = 10000$ and $\theta_2 = 2$.

- For all simulated schemes, the transmit power per bit, E_b, is standard: $E_b =$ 14[dBm]=25[mW]. All schemes utilize a standard DSSS with acquisition window length $F = 512$, at the physical layer. Thus the acquisition capacity of the system is approximately 36 users/nodes (please see Appendix A for acquisition capacity discussion). An exception here is the distributed transmit beam forming, which is itself a physical layer protocol. The evaluation of its performance in the simulation system is described with more detail later in the next subsection section.

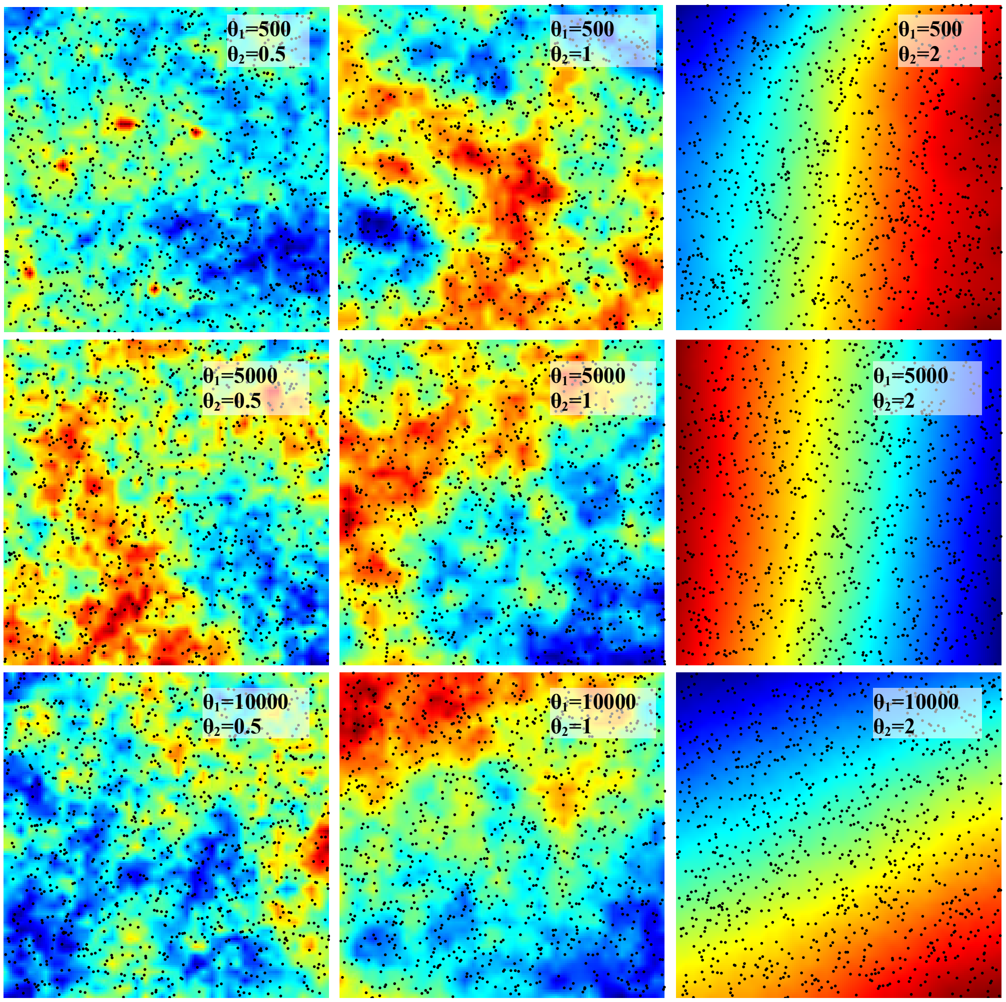


Figure 2.8: Influence of parameters θ_1 and θ_2 in the correlation model in (2.28) on the spatial distribution of phenomenon values. Notice that θ_2 affects significantly the behavior of the phenomenon. Nodes are placed uniformly at random. Network density 2.5 nodes/m^2.

2.4.3.2 Simulated Schemes

The simulation results below qualify the energy efficiency of the system, the distortion of the sink measurement, and compare ES performance to schemes communicating via

- distributed source coding (DSC) (e.g. [14], [13]);

- non-cooperative transmission with decorrelation (NCD) ([21]);

- cooperative distributed transmit beamforming (BF) ([34]).

- Per ES, the range of phenomenon values is quantized into a sequence of intervals where $M = 5000$. Each interval is of length $\epsilon = 1$, with $\sigma_W = 0.25$.

- Per NCD, k representative nodes are selected by the CC-MAC algorithm as described above, and each of the k nodes transmits, non-cooperatively, the entire measurement. The transmissions of highly correlated nodes are eliminated with the goal of better energy efficiency. The scheme's energy efficiency is equivalent to a duty cycle scheme. In the latter, only one node is selected to transmit the entire measurement from each of the k representative groups, at each timeslot.

- The algorithm underlying the DSC scheme is a single hop variation of Algorithm 2 in [13] and Algorithm 1 in [14]. The algorithm is frequently used in contemporary works on DSC in wireless sensor and ad-hoc networks. Given a neighborhood radius r_i around a representative node i, the algorithm constructs an ordered sequence C_i of the nodes that are within distance r_i of i. Next, in the order of the sequence, the algorithm allocates rates as follows: the first node in the sequence is allocated rate $R_1 = H(X_1)$; the second is allocated $R_2 = H(X_2|X_1)$, etc.; the last node in the neighborhood is allocated rate: $R_K = H(X_K|X_{K-1}, \ldots, X_1)$, assuming $|C_i| = K$; $r_i = r_{corr}$ around each representative node.

- The BF scheme evaluated here is based on the cooperative distributed transmit beamforming scheme suggested in [34]. An assumption in [34] is that upon each transmission, cooperating nodes are synchronized in frequency with each other. Furthermore, they need to be synchronized in phase so that the gains of beam forming are realized due to constructive interference at the sink. Satisfying the latter assumption, a simple randomized algorithm is offered in [34] to achieve phase coherency of the transmitted bits at the sink. Similarly to the above schemes, the beamforming groups G_i are formed within correlation radius r_{corr} around each of the k representative nodes.

2.4.3.3 System Capacity and Error Rates.

Since each node sending a bit per ES needs to be identified by the sink, we consider a regular DSSS physical layer as outlined in Fig. 2.4A above. Typically, a DSSS communication system is described both by *post-acquisition-based* capacity and *acquisition-based capacity*. The *post-acquisition-based* capacity of the system characterizes the maximum number of nodes in a system so that, given processing gain F (number of PN sequence chips per bit), the probability of a bit error is less than a threshold ϵ. The acquisition-based capacity of the system characterizes the maximum number of nodes in a system so that the probability p_a of acquiring wrong PN sequence at the sink is less than a threshold ϵ.

The Encoded Sensing (ES) schemes rely only on the correctness of the output of the matched-filters at the sink after the acquisition stage. Hence, we are interested in the acquisition-based capacity of the system. ES requires that the number of nodes $|G|$ is less than or equal to the acquisition-based capacity of the system. Assume that the signals of concurrently active nodes in set A_x arrive at the sink with independent random time-delays (within any given reporting time-slot) and with independent random carrier phases. Note that if more timing information is available a priori, for example by virtue of a side feedback channel devoid of interference, the probability of acquiring wrong PN sequence would be lower and the system capacity would be larger. This would be a viable approach in some practical scenarios, but we do not implement this approach in our simulation system. It is well known (e.g. see [107]) that the probability,

p_a, of acquiring a wrong PN sequence in our setting is given by

$$p_a \approx \frac{F-1}{2} e^{-\frac{3F}{4(|G|-1)}} \tag{2.29}$$

Notice that the system is interference limited. In contrast, per the NCD scheme, a single representative node from group G transmits, sending $\log(M)$ bits. The bit error rate of the NCD scheme utilizing, for instance, binary amplitude shift keying OFDMA over AWGN channel is given by

$$p_b \approx \frac{1}{2} erfc\left(\sqrt{SNR}\right) \tag{2.30}$$

The bit error rate expression for p_b in (2.30) remains the same for BPSK and single user DSSS systems.

In the simulation results presented here, when comparing the energy performance of the NCD and ES schemes we are conservative and require not only that w.h.p. $p_a < p_b$, but also that $P_a = 1 - (1 - p_a)^m < p_b$, where $m = |A_x|$. That is, we ensure that the probability P_a of not acquiring all m signals correctly is less than the probability of error of traditionally demodulating a single bit. Thus, the reduced energy consumption of the ES schemes is not an artifact of increased error rate vis-á-vis traditional modulation. More specifically, fig. 2.9 shows the bit error rate, p_b, of the NCD system, the probability p_a of acquiring a wrong PN sequence, and the probability P_a that at least one of the received $|A_x|$ PN waveforms was acquired incorrectly per the ES schemes, for different group sizes $|G|$. Notice that for $F = 512$ in all cases $p_a < P_a < p_b$. Here SNR $\approx$ 5.5dB and $p_b \approx 0.005$ from (2.30). For completion, we also include an analysis of the system post-acquisition-based and acquisition-based capacities. Using (2.29) the acquisition-based capacity, C_{acq}, of the system is given by

$$C_{acq} \approx \frac{3F}{4ln\left(\frac{F}{2\epsilon}\right)} \tag{2.31}$$

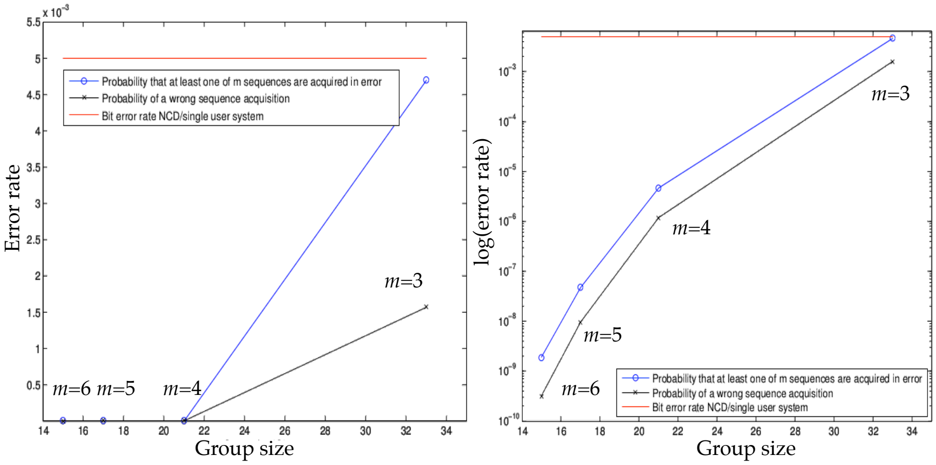


Figure 2.9: Probability of wrong PN sequence acquisition at the sink (both in terms of P_a and p_a) and bit error rate p_b for NCD (i.e. single representative node transmissions), for varying number of nodes in a group. Notice that in our simulation, for all group sizes $|G|$, we require and satisfy $p_a < P_a < p_b$.

It is well known (e.g. [107]) that the probability of bit error p_err in a interference-limited DSSS system with $|G|$ users, under Gaussian approximation, is given by

$$p_{err} \approx Q\left(\sqrt{\frac{3F}{|G|-1}}\right) \tag{2.32}$$

Let $\epsilon = 0.005$. That is, we require that $p_{err} < \epsilon = 0.005$. When $F = 512$, using (2.32) the post-acquisition-based capacity of the system is 230 nodes; in contrast, using (2.31) we obtain $C_{acq} = 35$. In this case, the system capacity is determined by acquisition performance. In all our simulation runs we require and satisfy $C_{acq} = 35 > |G|$.

2.4.3.4 Energy Efficiency and Estimate Inaccuracy

All schemes are simulated and compared in terms of two metrics: *energy consumption* and *estimate inaccuracy* at the sink. Energy consumption is given by the average energy consumption E per node in the network for reporting rounds. Estimate inaccuracy is given by the average inaccuracy of the phenomenon point source estimate at the sink over τ rounds:

$$\bar{V} = \frac{1}{\tau}\sum_{i=1}^{\tau} V_i \tag{2.33}$$

where at round i, $V_i = 100|s_i - s'_i|/M$. s_i is the true value of $\mathbf{S}$ at (x_i, y_i), and s'_i is the estimated measurement at the sink given the report of representative group G_i (or in the case of NCD the representative node i).

The average energy consumption per network node E for ES tends to be less than that of NCD and DSC as shown on fig. 2.10 and comparable to the performance of distributed beamforming with $K = 5$ cooperative beamforming nodes. As expected, both ES-MCDE and ES-C perform identically in terms of energy efficiency. When more nodes are added to the network (in effect increasing its density), as expected E decreases for all schemes. However, both ES schemes benefit substantially more from the network density increase, compared to DSC and NCD. Larger network density allows for smaller values of K^* leading to a lower number of transmitting nodes in $|A_x|$ required to convey measurement x. At network densities of 1.5 nodes/m^2, ES consumes 2 times less the transmission energy of NCD; at 1.9 nodes/m^2, the energy is 3 times less; and nearly 5 times less at densities around 3.5 nodes/m^2 depending on the correlation radii. $|G_i|$ grows from 15 to 33 nodes as density increases (notice that $|G_i|$ is always under acquisition capacity) and $|A_x|$ decreases from 7 to 3 nodes. BF also benefits from the addition of cooperative nodes. Evaluating the performance of BF, the phase synchronization feedback algorithm is successfully run at each beamforming node to achieve 70% phase coherency at the receiver (which is equivalent to achieving about 80% of beamforming gains). Operating below its theoretical efficiency, BF outperforms the ES for the case of 10 cooperating nodes in the

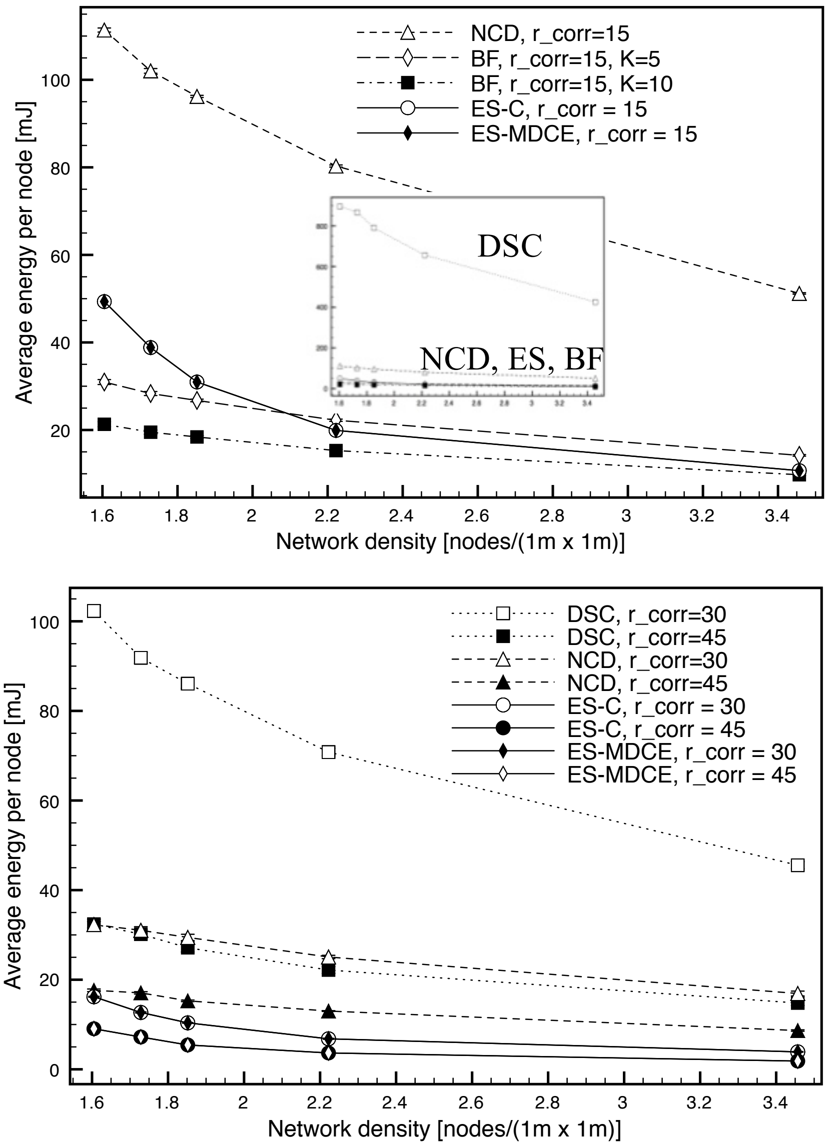


Figure 2.10: Energy consumption for 150 rounds and different r_{corr} values. **Top**: r_{corr} = 15[m]; **inset**: DSC performance; **bottom** r_{corr} = 30[m] and r_{corr} = 45[m]. Energy consumption decreases as the correlation radius increases (number of representative nodes decreases). The performance of ES-MDCE is identical to the performance of the basic ES-C encoding scheme.

region of lower network densities. However, the synchronization assumptions of ES are much more relaxed than those of the BF scheme. In reality, synchronizing more than 10 nodes would pose a challenge to any practical distributed transmit beamforming system.

Interestingly, DSC compares rather poorly with the rest of the schemes in terms of energy efficiency. This is so since all nodes within a given neighborhood with radius r_{corr} transmit with positive rates, allocated to account for nodes observations' correlation. The total rate equals at least the joint entropy of the observed measurements and is higher than the rest of the schemes.

Figure 2.11 demonstrates that the inaccuracies in the source estimates are rather similar for ES, NCD, and BF. This is expected in the case of ES and NCD since as shown in Appendix B the distortion of the two schemes numerically converges. All four schemes report information only about the value of the phenomenon at the representative nodes i. However, ES-MCDE outperforms slightly ES-C, achieving lower estimate inaccuracy. DSC obtains slightly lower inaccuracy, since all nodes in each r_{corr} neighborhood around a representative node send information regarding the sensed value at their position.

2.5 Number of Transmissions and Energy Efficiency

This section analyzes the minimum required number of nodes in G, required for the ES operation. The arguments below are valid for both ES-C and ES-MCDE.

In (2.3) it is required that $\binom{|G|}{K} \geq M$, where $K = |A_x|$. If the group size $|G|$ is fixed, $\binom{|G|}{K}$ is maximized for $K = |G|/2$. What is the least value of $|G|$ that

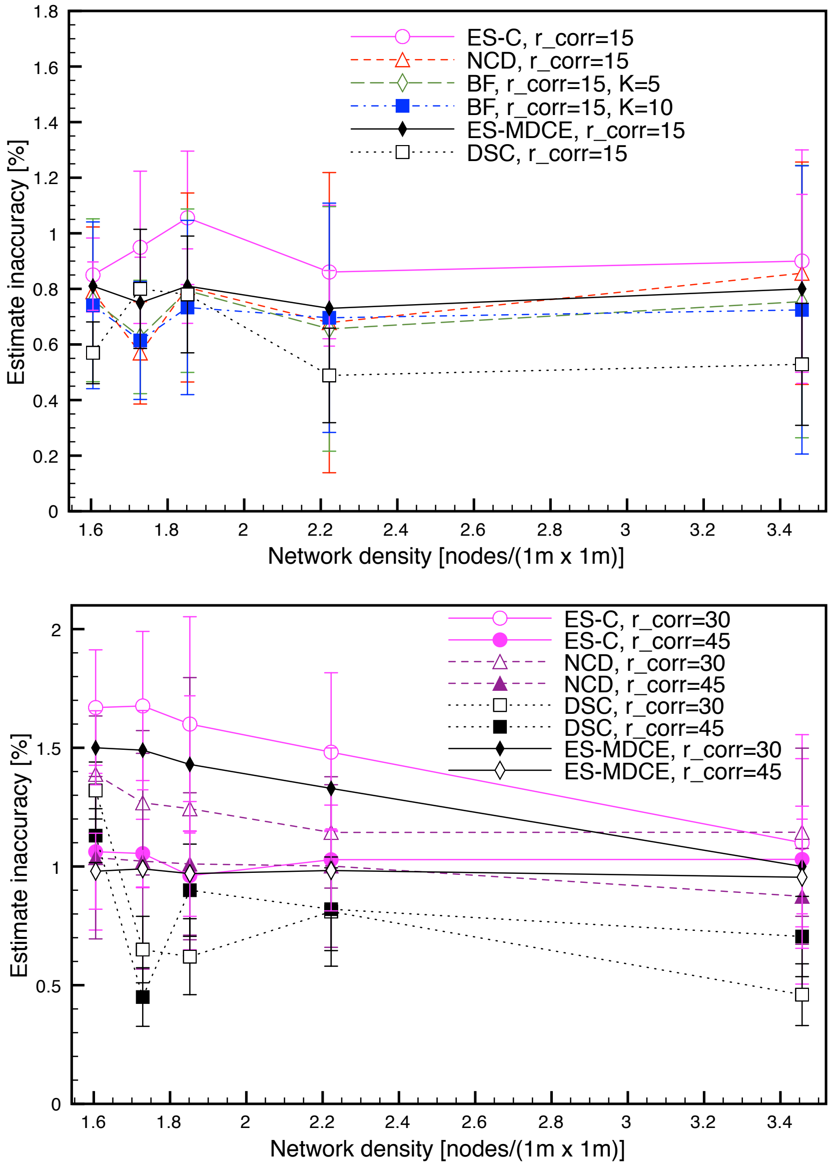


Figure 2.11: Estimate inaccuracy. **Top** $r_{corr} = 15$[m]; **bottom** $r_{corr} = 30$[m] and $r_{corr} = 45$[m]. ES-MDCE achieves lower inaccuracy and lower variance than the basic ES-C encoding scheme. However, all four scheme estimate the source relatively accurately. The measurements at the representative sites found via vector quantization model represent well the phenomenon.

guarantees $\binom{|G|}{|G|/2} \geq M$? Using Sterling's approximation yields

$$\binom{|G|}{|G|/2} \geq 2^{|G|-1} / \sqrt{|G|/2}$$

Therefore, $2^{|G|-1} / \sqrt{|G|/2} \geq M$ implies $\binom{|G|}{|G|/2} \geq M$. Solving for $|G|$:

$$0.5|G| \geq log\left(0.5|G|\right)/4 + log\left(M\right)/2 + 0.5$$

Also notice that $log\left(0.5|G|\right)/4 \leq 1$ for practical values of $|G|$ (up to $|G| = 32$ nodes). Then,

$$|G| \geq 2\lceil log\left(M\right)/2 + 1.5\rceil \tag{2.34}$$

Satisfying (2.34) ensures there are enough codewords for each message in M. Here, $K^* =|G|/2$. This is the maximum possible number of nodes in the set A_x, largest codeword weight, and the worst case of ES operation. Allowing larger group sizes $|G|$ substantially reduces the value of K^* utilized in ES as shown in fig. 2.12. In practice, $K^* << |G|/2$ is sufficient to encode M messages, as observed in the example WSN application of the previous section. One can now compare the worst-case energy efficiency of ES vs. the average energy efficiency of NCD. Since each node sending a bit per ES needs to be identified by the sink, consider a DSSS physical layer is utilized as outlined above. I.e. the energy E_b consumed per bit transmission is spread over a PN sequence. In the case of ES, we do not need to detect the sign of the signal (i.e. we are not interested whether the transmitted bit is 0 or 1 as long as some signal is there). Suppose $K^* = |G|/2 = \lceil log\left(M\right)/2 + 1.5\rceil = |c_x|$ bits are transmitted per message, as in the worst case of ES operation from (2.34).

In contrast, the NCD scheme over the same underlying DSSS physical layer would require transmission of a single representative node from group G, sending $log(M)$ bits. Each of these bits, as standard in single user DSSS systems, is

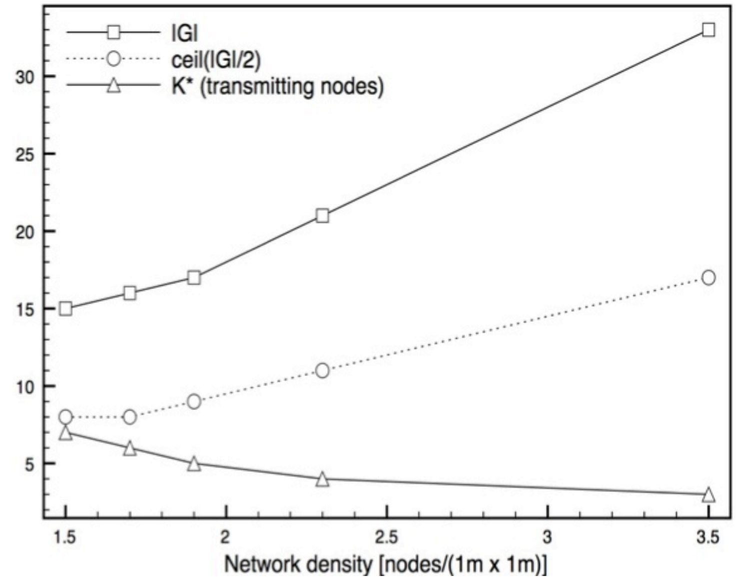


Figure 2.12: Number of nodes in a group G_i and number of transmitting nodes K^* for different network densities. Even at lower densities, K^* is less than $|G|/2$, the worst-case of ES operation, and drops further with density. K^* is the same for both ES-MDCE and ES-C.

spread over a PN sequence and modulated antipodally. The sign of each bit, 0 or 1, *must be detected*.

Given a transmission energy budget of E each, ES and the NCD schemes can send respectively d_{ES} and d_{NCD} messages in total: $d_{ES} = \frac{E}{(|c_x|E_b)}$ and $d_{NCD} = \frac{E}{log(M)E_b}$. In the worst-case for ES, $|c_x| = \lceil log\,(M)\,/2 + 1.5\rceil$ and

$$d_{ES} = \frac{2d_{NCD}}{(1 + 3/log(M))} \tag{2.35}$$

Thus, ES in its worst regime of operation is close to 2 times as energy efficient as NCD or a corresponding duty-cycling scheme. Respectively, the ES scheme achieves at least a factor of 2 better energy efficiency than NCD in simulations. Suppose the NCD scheme utilizes different, for instance non-antipodal, modulation so that in effect only $\lceil log(M)/2\rceil$ signals are transmitted. Still, the energy per bit required to achieve the bit error rate obtained via the former antipodal modulation would need to at least double (for instance, if orthogonal instead of antipodal modulation is used). It is well known that a DSSS system achieves the same bit error as a non-spread spectrum system with the same modulation format. On an AWGN channel (without interference), the unspread scheme would require the same amount of energy as a DSSS-based non-cooperative scheme.

2.6 Other Collaborative Transmission Systems

A number of different schemes that take advantage of data correlation and "approximately common data" have been suggested in the technical literature.

One of the first and most prominent examples of collaboration across multiple wireless sensor nodes is cooperative communication, discussed in [108]. Inspired by the gains of spatial diversity realized in cooperative MIMO systems the authors of [109], study cooperative communication in the context of WSN comprising sets of nodes where each node is equipped with a only single antenna. The nodes within each set transmit cooperatively (portions of) the same message to a sink. Since different nodes have different spatial coordinates, the sink receives multiple instances of the same message that have suffered fading over statistically independent spatial paths. The earliest realizations of cooperative transmission utilize the *decode-and-forward* [110] and the *amplify-and-forward* mechanisms [111]. Various schemes for optimal selection of cooperating nodes (e.g. [112]), algorithms for cooperative space-time coding of the transmitted messages (e.g. [113]); and cooperative game theoretic frameworks optimizing network quality of service per given energy budget ([114]) have been further investigated. Encoded sensing is distinct from such cooperative communication strategies. Per ES, each node in the subset A_x of G transmits a single bit directly to the sink, without additional communication across collaborating nodes.

ES does not rely on spatial diversity in order to achieve energy efficient communication. This is in contrast to more recent distributed transmit beam forming systems (BF) [34]. Per BF, a set of cooperating nodes simultaneously transmit copies of the same message to a single sink. The signals of the cooperating nodes are specifically synchronized in phase and frequency resulting in constructive interference at the sink. Theoretically, employing n cooperative beamforming nodes could lead up to a factor of n energy savings compared to non-cooperative signaling. However, in the most recent proof-of-concept beamforming systems, noiseless 1-bit feedback from the sink is assumed so that all cooperatively transmitting nodes are approximately synchronized in phase. This feedback is provided via a separate cable connection (e.g. [115]) to the sink; this is often not feasible in practice. ES does not require as fine synchronization of message transmissions as beamforming. Furthermore, ES continues to operate even if common data is not present (i.e. b in (2.1) is 1).

A somewhat separate stream of technical literature discusses the collaborative operation of nodes in WSN taking into account correlations in measurements. For example, a scheme is introduced in [21] eliminating the transmission of measurements that are highly correlated. As shown in [21], the elimination of certain highly correlated messages would not distort (within a bound) the overall phenomenon estimate at the sink. Thus network energy is conserved. The outcome of the scheme is that within a group of nodes measuring the same (or very similar) values, only a single node transmits an entire measurement value at every time-slot.

In a similar spirit, techniques relying on distributed source coding have been suggested for WSN. The authors of [13] and [14] leverage results based on the Slepian-Wolf source-coding theorem. Namely, correlated sources can be compressed separately (e.g. without communication between the source nodes) and without loss to the level of their joint entropy. Consequently, according to a prespecified scheme, each sensor node in the network transmits fewer bits the more correlated its measurement is to reference nodes. As discussed in section 2.4, ES leverages the representative nodes selection algorithm in [21] and achieves more energy efficient data reports than distributed source coding.

The work of [116] provides a comprehensive overview and classification of various data sampling and compression schemes based on compressive sensing for WSN, some of which were outlined in subsections 1.3.1 and 1.3.2 of chapter 1. Notice that in itself a number of these compressive sensing methods are orthogonal to techniques utilizing spatial correlation across nodes and respectively encoded sensing. In the context of ES, for instance, the nodes within each representative group would have very highly correlated intra-signal structure (since these nodes are spatially close). Exploiting the K-sparsity of this structure via compressive sensing, the nodes within each representative group could access the same set of $O(Klog(n))$ incoherent measurements over n timeslots. ES can then be utilized to transmit each one of the $O(Klog(n))$ samples in each representative group. The extension of ES within this setting provides an interesting avenue for research and holds potentially substantial energy savings in various applications. In itself, ES does not require inherent K-sparsity in the observed phenomenon signal. The linear combination of signature waveforms characterizing the received ES signals at the sink at a given timeslot is guaranteed to be $|A_x|$-sparse. As the size $|G|$ of the representative group increases, $|A_x|$ decreases independent of the sparsity of the phenomenon signal.

CHAPTER 3

OPTIMIZING THE STRUCTURAL LEVEL

An important factor driving the number of packet re-transmission is network's links quality ([58], [59]). Among others, presence of obstacles between nodes; increasing interference as the density of nodes grows; and separation distance between wireless devices may all influence links' quality. The relative impact of each of these factors on network performance depends on the particular structural level and network deployment application.

For instance, the main reason for poor link quality in sparsely deployed outdoor sensor networks is the large separation distance between sensing nodes causing low SNR, and low packet reception rate (PRR) ([59]). In these settings, to improve links' quality and decrease communication cost in wireless networks, researchers and practitioners often rely on the deployment of *relay nodes*. Relay nodes do not introduce new traffic in the network and only retransmit the packets received from a set of source nodes.

Generally, given a set of *n* fixed source nodes, along with their locations and traffic demands, and a budget of *k* relay nodes, the problem posed is to find the optimal locations of the *k* relay nodes minimizing the communication cost in the network. This problem arises in many basic application scenarios. Pre-existing sensor networks comprising *n* nodes may contain "low quality" links incurring a large number of packet retransmission. For instance, the clusters positions picked by ES in the previous chapter may be far apart due to the data level model. Moving the *n* sensors may not be feasible from the perspective of sensor network deployers (e.g. [45]); or may diminish the fidelity of data received at the sink. In the case of ES for instance, representative groups positions (see definition (2.1) in the previous chapter) are picked to optimize the distortion of the data received at the sink. Changing these positions would increase this distortion. A solution could be the deployment of *k* new relay nodes.

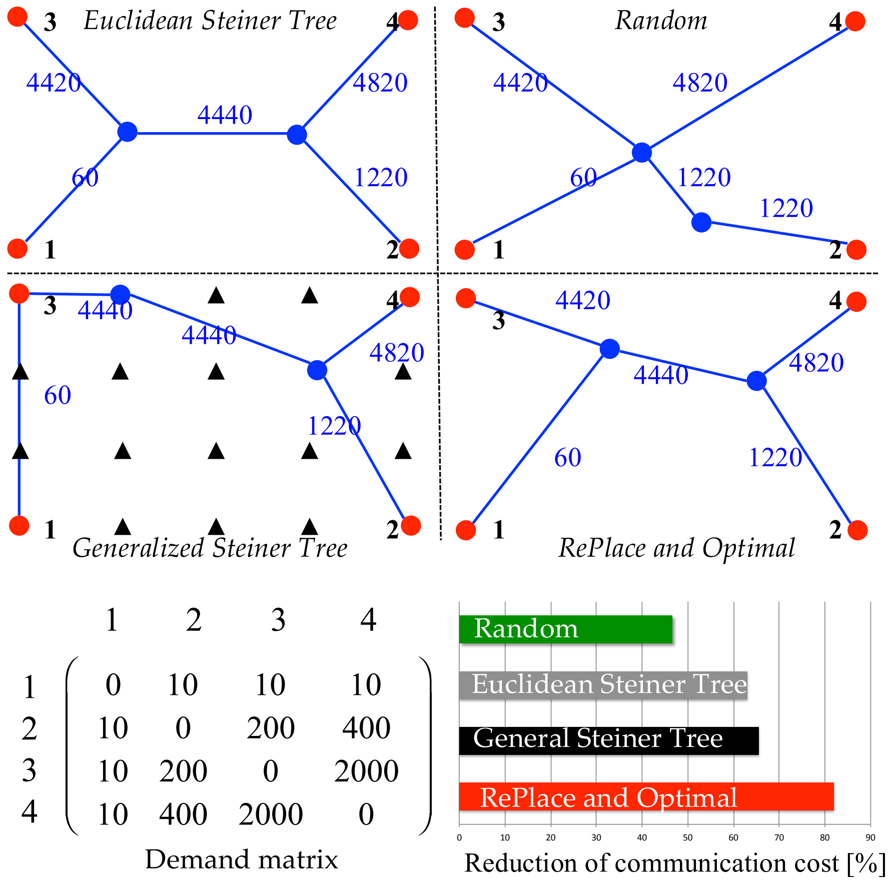
In other cases, even if link quality is uniform throughout a network of *n* devices, certain subsets of the devices may be more active, generating larger amounts of source traffic in the network. A subset of links may be more loaded, in effect raising the number of dropped packets on these links. Placing *k* relay nodes appropriately reduces the communication cost[1] in such scenarios.

3.1 Relay Placement Challenges

As shown in fig. 3.1, the appropriate placement of relay nodes may effectively reduce the length of the longest and more loaded links between source-destination pairs in a network, and thus potentially increases links' PRR and decrease communication cost. Even in the simple case of fig. 3.1, the optimal placement of relay nodes accounts for more than 80% reduction in communication cost. It is well known ([117]) that even random placement of relay nodes may theoretically have beneficial effects on network capacity as the network scales. However, in general the number of randomly placed relay nodes required to achieve noticeable impact on network performance is rather large. The random deployment of relay nodes is expensive and not practical. In the sample topology of fig. 3.1, the random placement of relay nodes achieves only about 45% in communication cost reduction.

Aside from the optimal and random placement, fig. 3.1 illustrates the relay nodes topologies generated by two other algorithms typically utilized in

[1] Communication cost is more rigorously defined in section 3.2.1.2.



$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 10 & 10 & 10 \\ 2 & 10 & 0 & 200 & 400 \\ 3 & 10 & 200 & 0 & 2000 \\ 4 & 10 & 400 & 2000 & 0 \end{array}$$

Figure 3.1: A sample network topology of $n = 4$ fixed nodes and a demand matrix $\mathbf{W}$ summarizing the traffic demands for different source destination pairs. There are $k = 2$ relay nodes to place in the network. The solutions obtained by the **Euclidean** and **Generalized Steiner** trees algorithms are not optimal in terms of reducing communication cost. The optimal solution is characterized in this chapter. The suggested **RePlace** algorithm (3.1) achieves it in this network sample.

various studies and applications requiring relay nodes deployment. The first algorithm solves the Euclidean Steiner tree problem and the second solves the General Steiner tree problem on the given sample topology. These algorithms however *do not truly solve* the optimal relay placement problem. The inefficiencies of these algorithms' outputs compared to the optimal relay placement stem from somewhat subtle but fundamental difference between the corresponding problem models.

The Euclidean Steiner tree consists of locations and links that interconnect the n given fixed nodes in the plane. Each connecting link has an associated weight equal to the Euclidean distance between its vertices. The goal is to pick the locations on the plane that will minimize the sum weight of the interconnecting links. The relay nodes are placed at these locations. The problem is NP-complete ([83]), however good approximations are efficiently found. Since the sum of distances between the relay nodes is minimized, the intuition to such a solution is that the overall links' quality in the network increases, driving the number of retransmitted packets and communication costs down. While such approaches are frequently used in practice (e.g. [59] in the field of sensor networks; and [118] in the field of robotics), they are not necessarily optimal, as fig. 3.1 demonstrates. First, the Euclidean Steiner tree does not account for the traffic loads on links: heavily utilized links may require more relay nodes placed closer to them. For instance, in fig. 3.1, the traffic between sources 3 and 4 is significantly larger than the rest of the links, shifting the optimal positions of the relay nodes away from the Euclidean Steiner tree. Second, the SNR and respectively the number of packet retransmissions due to poor PRR do not depend linearly on the distance between receiver and transmitter.

These issues seem to be remedied by modeling the relay placement problem as an instance of the General Steiner tree model. In the General Steiner tree problem, the input consists of a graph $G(V, E)$, where each edge $ij \in E$ has an associated cost c_{ij}. Given a set of terminal nodes $N \subseteq V, |N| = n$, the goal is to find a tree ST of minimum cost spanning the vertices in N. Here, the weights of the edges are arbitrary. As part of their influential work ([58]), Krause et al. utlize the General Steiner tree model to suggest locations for communication nodes within a network of wireless sensors deployed to maximize the amount of gathered sensing information and minimize communication cost in the network. In [58], V is finite and represents the potential locations of network nodes. Each edge $ij \in E$ in G has weight equal to the expected number of times a packet needs to be retransmitted by i so that the packet is received successfully by j. In this model, the communication cost depends on the locations of the nodes and distance between them but, unlike the case of the Euclidean Steiner tree, not necessarily linearly. The communication nodes are placed at the Steiner tree vertices in ST spanning the set of sensor nodes.

The General Steiner tree model for placing relay nodes appears sound, however it omits a few important factors affecting the optimal placement of relay nodes. First, the ideal locations of the relay nodes depend on how the routing in the network is constructed: the communication cost on a link $ij \in E$ depends on the amount of traffic flow on ij. The more the packets flowing on link ij are, the greater is the expected number of retransmissions, i.e. communication cost, on link ij. Furthermore, the ideal routing in the network depends on the positions of the relay nodes: one could obtain a better routing if one were to position the relay nodes elsewhere within a set of available locations. This hints that the locations of the relay nodes and the routing should be optimized simultaneously *working both on the structural and the procedural network levels*. The General Steiner tree approach to modeling the optimal relay placement does not capture these aspects of the problem. This is corroborated by the example in fig. 3.1, where Steiner tree based solutions do not match the optimal solution, achieving markedly lower reduction in communication costs even in this very simple network scenario. Furthermore, in practice, the relay nodes may occupy a continuum of points in the plane. In some cases, confining the points to a prespecified set of discrete locations as per the General Steiner tree model may not be feasible. To obtain a good approximation of the best relay nodes positions in network where n nodes span large area, one may need a large amount of possible locations in V, which increases significantly the complexity of the respective General Steiner tree approximation algorithms.

This chapter revisits anew the problem of placing optimally a set of k relay nodes within a network of n fixed nodes, with the goal of reducing network communication cost. This task is formulated as a novel optimization problem that generalizes previously studied wireless network node placement problems.

- The presented optimization framework allows nodes to be placed at a *continuum of points* on the plane [2]. Furthermore, the relay nodes' *locations and routing traffic patterns* in the network are simultaneously optimized.

- Via reduction to a graph clique problem, it is shown that per the optimal relay placement model even listing a set of feasible sites for the placement of relay nodes is NP-hard.

- Exploiting convexity in a special case of the network communication cost function, an optimal algorithm solving the relay placement problem and minimizing network overhead retransmissions is described. The optimal algorithm, however, is exponential on the number of nodes in the network and hence not practical for larger network instances.

- An efficient relay placement and routing heuristic algorithm (**RePlace**) is suggested. Numerically it is shown that **RePlace** outputs optimal or very close to

[2] We do not consider here spatial restriction on the possible locations of the relay nodes in the plane (e.g. due to physical obstacles such as ponds or rivers), however the framework generalizes to settings, where relay nodes may only be placed within constraints of continuous convex sets of points.

the optimal solutions (within 2-3% of the optimal communication cost) in small network instances. **RePlace** is implemented in the full network stack simulator JiST/SWANS ([119]), where practical network effects (e.g. link asymmetry, interference, noise, collisions, etc.) are present. As the number of relay nodes increases **RePlace** eliminates almost entirely overhead communication costs, in terms of packet retransmissions due to low quality links. Furthermore, the resulting delay in networks with relay nodes placed according to the **RePlace** solution, decreases down to about 35% of the delay achieved in the same networks without placement of relay nodes.

- Finally **RePlace** replace is compared to alternative relay placement schemes in the recent technical literature.

3.2 Optimal Topology for Relays in Wireless Networks

3.2.1 System Model

3.2.1.1 Link model

Let *ij* denote the wireless link between two network nodes *i* and *j*. We calculate the packet error rate r_{ij} on link ij, assuming the log-distance path loss model ([120]). This wireless signal propagation model has been known to realistically characterize a number of low power network deployment scenarios (e.g. see [121] and [59]). Assuming *i*'s transmit power is P_t, the received power P_r at *j* is given by

$$P_r = P_t - a(d_0) - 10\alpha \log_{10}(d_{ij}/d_0) + \eta(0, \sigma) + \chi(0, \sigma_1) \tag{3.1}$$

and the SNR at j is correspondingly

$$\gamma(d_{ij}) = P_t - a(d_0) - 10\alpha \log_{10}(d_{ij}/d_0) - \eta(0, \sigma) - \chi(0, \sigma_1) \tag{3.2}$$

where $a(d_0)$ is the attenuation at reference distance d_0. $\eta(0,)$ and $\chi(0, \sigma_1)$ are normal random variables modeling the thermal noise power and shadowing respectively.

Per the physical reception model in Gupta and Kumar's seminal paper ([117]), the received SNR has to be greater than a minimum threshold ϕ for a successful transmission:

$$\gamma(d_{ij}) \geq \phi \tag{3.3}$$

At present, we assume that there is no interference from nodes' transmissions. In section 3.2.5.2, we study the effect of interference. Hence, the bit error rate p_{ij} on link ij is given by

$$p_{ij} = Q\left\{\left[2\gamma(d_{ij})\right]^{\frac{1}{2}}\right\} \tag{3.4}$$

if BPSK modulation is utilized. The results below are easily extensible for other commonly utilized modulation schemes.

The packet error rate r_{ij} on ij is then given by

$$r_{ij} = 1 - (1 - p_{ij})^b \tag{3.5}$$

where b is the packet length in bits.

Physical link cost: the cost c_{ij} of link ij is defined as

$$c_{ij} = 1/(1 - r_{ij}) \tag{3.6}$$

c_{ij} accounts for the number of dropped packets due to low SNR at receiver j in the network. Notice that c_{ij} captures well links' communication cost. The physical link cost c_{ij} however does not account for the full communication cost on link ij, as discussed below.

3.2.1.2 Network model

Let V be a set of network nodes with equal computational and communication capabilities. We assume that links in the network are symmetric. The network is modeled as an undirected, connected, weighted graph $G(V, E)$. E is the set of wireless links. The weight of edge $ij \in E$ is c_{ij}. The graph G is captured by its weighted adjacency matrix $\mathbf{C} = \left[c_{ij}\right]_{|V|x|V|}$. Notice that the values c_{ij} depend on the distance between the nodes i and j, and hence on the positions of the nodes i and j. Link ij exists iff $d_{ij} \leq R$, where R is defined as follows.

Definition 3.1. Maximum Transmission Range R: *The maximum transmission range R* of node i is the maximum distance away from i at which, a fraction β of the time, the BER is less than $0.5 - \epsilon, \epsilon > 0$.

The parameters ϵ and β depend on QoS constraints.

Suppose the initial demand matrix $\mathbf{W} = [w_{sd}]_{|V|x|V|}$ is provided. Each entry w_{sd} of $\mathbf{W}$ captures bidirectional traffic demand [packets/sec] between nodes s and d: the sum of the traffic demands from node s to d and from node d to s. A demand pair is denoted $\langle sd \rangle$. For all given pairs $\langle sd \rangle$ we can find a routing path $y_{sd} \subset E$ connecting s with d. Let Y^{ij} be the set of all $\langle sd \rangle$, such that $ij \in y_{sd}$. The traffic on ij is then

$$q_{ij} = \sum_{\langle sd \rangle \in Y^{ij}} w_{sd} \tag{3.7}$$

Link ij is utilized if $q_{ij} > 0$.

Network communication cost: Assuming each packet error on ij is an independent event with probability r_{ij}, the expected number of packet retransmissions until q_{ij} packets are successfully received is given by

$$f_{ij} = q_{ij} c_{ij}. \tag{3.8}$$

The quantity f_{ij} is the communication cost on link ij. The greater the packet error rate r_{ij} on ij, the larger the physical cost c_{ij}. The larger the q_{ij} of link ij, the greater f_{ij} and the average packet delay on link ij. If link ij is not utilized it does not carry traffic: $q_{ij} = f_{ij} = 0$. If $p_{ij} > 0$, then $c_{ij} > 1$ and $f_{ij} > q_{ij}$. Ideally, if $p_{ij} = 0$, then $c_{ij} = 1$ and $f_{ij} = q_{ij}$.

The total *network communication cost* is given by

$$F = \sum_{ij \in E} f_{ij} \tag{3.9}$$

Intuitively, minimizing F would improve network performance in terms of network goodput and average packet delay. In the following sections, we study and quantify how the addition of relay nodes, so that F is minimized, impacts these network performance metrics.

3.2.2 Communication Cost Minimization

In this section, we formulate the general problem of communication cost minimization in wireless networks where a set of *relay nodes* is added to an initial set of *fixed nodes*. The relay nodes' positions are picked arbitrarily in the network area, and the positions of the fixed nodes cannot be changed. We describe an optimal algorithm for placement of the additional relay nodes that minimizes the communication cost in the resulting network topology.

Given a network of n fixed nodes and a set K of new relay nodes, where $k = |K|$, our task is the minimization of overall communication cost in G', where G' is the resulting network graph with vertex set $N \cup K$ and edge set E'. The relay nodes are not sources of traffic and can only offload the traffic from the original n nodes.

From (3.9), f_{ij}, $ij \in E'$, depends both on the distance between i and j (via the term c_{ij}) and the traffic routed through link ij (via the term q_{ij}). Hence, we need to jointly optimize two sets of variables to minimize communication cost:

(a) *the positions of relay nodes*; and

(b) *the routing* over the links in the relay network.

Notice also that

(c) *changing the positions of relay nodes* may affect the optimal routing paths in the network and *changing the routing paths* in the network in turn may affect the optimal positions of the relay nodes.

3.2.2.1 General Relay Placement Problem

Let $\mathbf{Q} = [q_{ij}]_{(n+k)(n+k)}$ denote the network traffic flow matrix, where q_{ij} denotes the traffic on link $ij \in E'$ as defined in (3.7). The coordinates of the k relay nodes are denoted by $(x_j, y_j), 1 \leq j \leq k$. We let $v = [(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)]$, $\mathbf{v} \in \mathbb{R}$. $\mathbf{C}(\mathbf{v})$ denotes the weighted adjacency matrix of G'. Its entries depend on $\mathbf{v}$. The general relay placement problem (GRPP) becomes

$$\min_{\mathbf{Q},\mathbf{v}} \{F(\mathbf{Q}, \mathbf{v})\} \equiv \min_{\mathbf{Q},\mathbf{v}} \{\mathbf{Q} : \mathbf{C}(\mathbf{v})\} = \min_{\mathbf{Q},\mathbf{v}} \left\{ \sum_{i=1}^{n+k-1} \sum_{j=i+1}^{n+k} f_{ij} \right\} \tag{3.10}$$

As usual, $\mathbf{A}$:$\mathbf{B}$ denotes the inner product of matrices $\mathbf{A}$ and $\mathbf{B}$; and we have

$$\min_{\mathbf{Q},\mathbf{v}} \left\{ \sum_{i=1}^{n+k-1} \sum_{j=i+1}^{n+k} f_{ij} \right\} = \sum_{i=1}^{n+k-1} \sum_{j=i+1}^{n+k} q_{ij} \left[1 - Q\left(\sqrt{2\gamma\left(d\left(x_i, y_i, x_j, y_j\right)\right)} \right) \right]^{-b} \tag{3.11}$$

$d(x_i, y_i, x_j, y_j)$ denotes the Euclidean distance between any two points i and j with coordinates $(x_i, y_i), (x_j, y_j)$.

The candidate solutions to the GRPP (3.10) are uncountably many since $\mathbf{v}$ may include any point on the plane. However, the different possible routing paths that satisfy (3.7) and hence the different matrices $\mathbf{Q}$ are finitely many. If we were able to compute the optimal vector $\mathbf{v}$ for each possible input $\mathbf{Q}$, we would have an optimal enumeration algorithm for solving the GRPP (3.10).

3.2.2.2 Relay Placement with Fixed Routing

Suppose the traffic matrix $\mathbf{Q}$ is provided. Can we find the optimal locations (i.e. $\mathbf{v}$) of the relay nodes and solve

$$\min_{\mathbf{v}} \left\{ F^{\mathbf{Q}}(\mathbf{v}) \right\} = \min_{\mathbf{v}} \left\{ \sum_{i=1}^{n+k-1} \sum_{j=i+1}^{n+k} q_{ij} \left[1 - Q\left(\sqrt{2\gamma \left(d\left(x_i, y_i, x_j, y_j \right) \right)} \right) \right]^{-b} \right\} \tag{3.12}$$

where q_{ij}'s are no longer optimization variables but the entries of the given matrix $\mathbf{Q}$? We label this problem as the Relay Placement with Fixed Traffic (RPFT).

Note that RPFT's solution must satisfy a set of simple geometrical constraints due to the properties of wireless links. Showing that the function $F^{\mathbf{Q}}(\mathbf{v})$ is convex under these constraints, we are able to provide an optimal solution to the RPFT. This allows us to pose GRPP as a combinatorial problem.

- Convexity of RPFT:

Let $\mathbf{v} = (x_1, y_1, x_2, y_2, \ldots, x_k, y_k)$ be the set of variables representing the coordinates of the k relay nodes. We begin by noting that x_i and y_i are independent: changing x_i or y_i does not change x_j or y_j, for any i and j. Next, we recall the following theorem.

Theorem 3.1. *The function*

$$F^{\mathbf{Q}}(\mathbf{v}) = \sum_{i=1}^{n+k-1} \sum_{j=i+1}^{n+k} q_{ij} g\left[d(x_i, y_i, x_j, y_j)\right] \tag{3.13}$$

is convex if $g(z) : \mathbb{R}^+ \to \mathbb{R}$ *is convex and non-decreasing.*

Proof. Please refer to [122], p. 434. □

Consider the function $F^{\mathbf{Q}}(\mathbf{v})$ in (3.12). Let

$$g\left[d(x_i, y_i, x_j, y_j)\right] = \left[1 - Q\left(\sqrt{2\gamma\left(d\left(x_i, y_i, x_j, y_j\right)\right)}\right)\right]^{-b} \tag{3.14}$$

We then can rewrite $F^{\mathbf{Q}}(\mathbf{v})$ as follows

$$F^{\mathbf{Q}}(\mathbf{v}) = const + \sum_{i=1}^{n+k-1} \sum_{j=i+1}^{n+k} q_{ij} g\left[d(x_i, y_i, x_j, y_j)\right] \tag{3.15}$$

To determine whether $F^{\mathbf{Q}}(\mathbf{v})$ is convex on some domain, by **Theorem** (3.1), we only need to determine whether $g(z)$ is convex and non-decreasing on that domain.

Observation 3.1. *The function* $g(z)$ *is convex and non-decreasing on the interval* $(0, rd_0)$*,* $\forall r$ *such that* $rd_0 > R$*.*

Observation 1 is analyzed with more detail in Appendix C. Figure 3.2 shows a plot of $g(z)$ when $P_t = 10[W]$, $R \approx 110[m]$, $b = 256[bits]$, $d_0 = 1[m]$. Given function $F^{\mathbf{Q}}(\mathbf{v})$ in (3.15) and $R \leq rd_0$, the RPFT problem becomes

$$\mathbf{v}' = \arg\min_{\mathbf{v}} \left\{F^{\mathbf{Q}}(\mathbf{v})\right\} \text{ s.t. } d(x_i, y_i, x_j, y_j) \leq R, \forall q_{ij} > 0 \tag{3.16}$$

The constraints inequalities in (3.16) are convex too (e.g. [122]).

Hence, for any fixed matrix **Q** and a network graph where links are constrained within transmission range R, we can solve (3.16) and find the positions

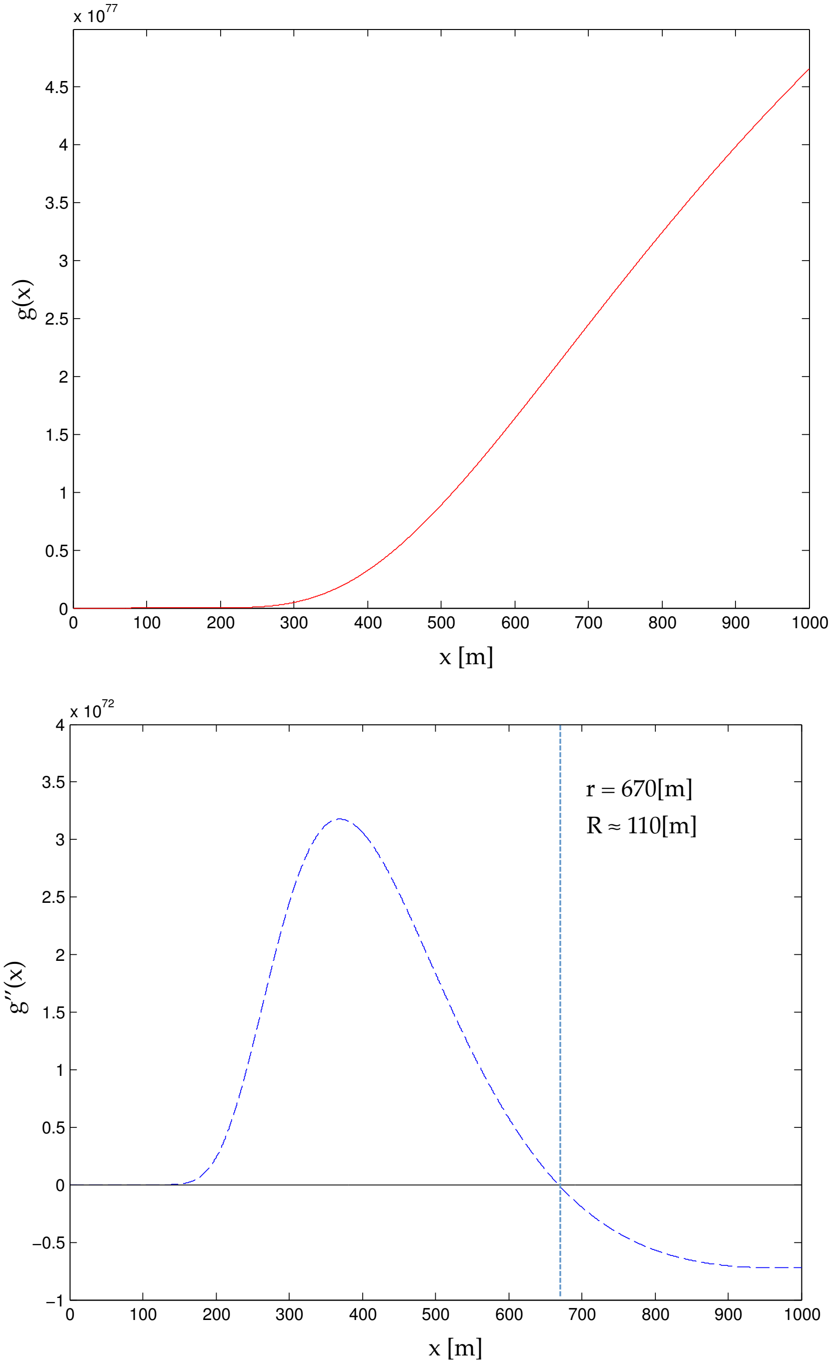


Figure 3.2: $g(z)$ behavior for $b = 256$[bits], $d_0 = 1$, $R = 110$[m]

of the relay nodes that minimize the network communication cost. Standard convex optimization algorithms such as the steepest gradient descent with constraints ([122]) could be utilized to solve the RPFT problem.

3.2.2.3 Optimal Brute Force Solution to the GRPP

Going back to the GRPP, we can now present a brute force combinatorial solution. Given a graph G of fixed nodes, its weighted adjacency matric C, demand matrix W, and a number k of relay nodes, the algorithm outputs as a solution the optimal coordinates $\mathbf{v}^*$ of the k relay nodes and the optimal routing Y^* in the resulting network G' on vertex set $N \cup K$. The pair $(\mathbf{v}^*, Y^*)$ minimizes the network communication cost.

Let Y be a set of routing paths on the vertices $N \cup K$ connecting all $\langle sd \rangle$ pairs. Let $\mathbf{Y}$ be the collection of all possible sets Y. $\mathbf{Y}$ has finite number of elements depending on n, k, and the maximum transmission radius R (assuming that the flow for each pair $\langle sd \rangle$ is routed on a single path).

The brute force algorithm (**BruteForceMin**) then follows:

1. for each set of routing paths $Y \in \mathbf{Y}$, generate the matrix $\mathbf{Q_Y} = [q_{ij}]_{(n+k)(n+k)}$ as in (3.7) and solve the constrained convex optimization problem (3.16) for $\mathbf{Q_Y}$, finding $\mathbf{v'}$ minimizing $F^{\mathbf{Q_Y}}(\mathbf{v}')$;
2. pick the solution $\mathbf{v}^*$ and the corresponding Y^* that yields the minimum $F^{\mathbf{Q_Y}}(\mathbf{v}')$ over all different routing paths $Y \in \mathbf{Y}$:

$$(\mathbf{v}^*, \mathbf{Y}^*) = \arg\min_{\mathbf{v}', Y} F^{\mathbf{Q_Y}}(\mathbf{v}') \tag{3.17}$$

The cardinality of $\mathbf{Y}$ is exponential of the number of vertices in $N \cup K$ and thus solving the GRPP using **BruteForceMin** is not practical for larger networks. In the next sections, we characterize the inherent computational hardness of the GRPP. We pose the GRPP as a maximization problem and provide an efficient, practical heuristic algorithm for it.

3.2.3 Zones, Overlaps and Feasible Polygons

3.2.3.1 Defining Zones and Overlaps

In this section, we define a set of geometrical constraints on the possible optimal positions of the relay nodes. This allows us to define a feasible set of polygons in the network area, where relay nodes can be initially placed. We show that listing these feasible polygons is an NP-hard problem.

Observation 3.2. *Given a link uv in G and a relay node i, the communication cost of uv can be reduced if and only if i is positioned within the lune formed by the overlap of the two circles with centers respectively u and v, each with radius d_{uv}.*

The lune of link uv is shown in fig. 3.3 (**top**) and is referred to as the *lune zone of link uv*. We can approximate the lune zone with a corresponding rhombus zone, as shown in fig. 3.3 (bottom). The rhombus approximation is chosen for ease of presentation. We can choose to approximate the lune with a different many-sided convex polygon. We denote this approximate rhombus zone of link uv by θ_{uv}, and refer to it as the zone of link uv. Let $\theta_{u'v'} \diamond \theta_{uv}$ denote the statement "the zone of link $u'v'$ overlaps with the zone of link uv". Let A be a set of links, such that $\theta_{u'v'} \diamond \theta_{uv}, \forall uv, u'v' \in A$. The polygon formed by the overlap of all zones of links in A is denoted by θ_A. We say $\theta_A > 0$ if the zones' overlap polygon has area greater than 0. Let $A \subseteq E$ be a set of links connecting some subset of fixed

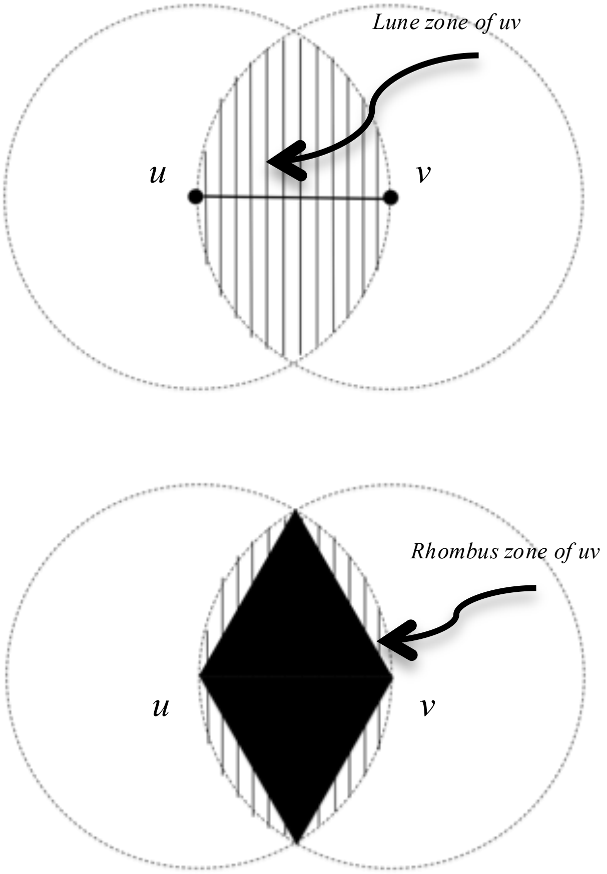


Figure 3.3: **Top**: Lune zone of link uv. **Bottom**: Rhombus zone of uv.

nodes in G.

Observation 3.3. *Given a set of links A, their zones, and a relay node i, i can reduce the communication cost of every link in A if and only if i is placed in θ_A, assuming $\theta_A > 0$.*

Observation 3 follows from *Observation* 2, since i is in the zone of every link in A, when placed within θ_A. Figure 3.4 demonstrates such an arrangement, where i is placed in $\theta_A, |A| = 3$. Consider the set $X(A) = \{\theta_S : S \subseteq A, \theta_S > 0\}$. In the example of fig. 3.4, $A = \{12, 13, 23\}$ and $X(A) = \{\theta_{\{12\}}, \theta_{\{13\}}, \theta_{\{23\}}, \theta_{\{12,23\}}, \theta_{\{12,13\}}, \theta_{\{13,23\}}, \theta_{\{12,13,23\}}\}$. Notice that a zone overlaps with itself. According to *Observation* 3, the relay node i can only reduce the communi-

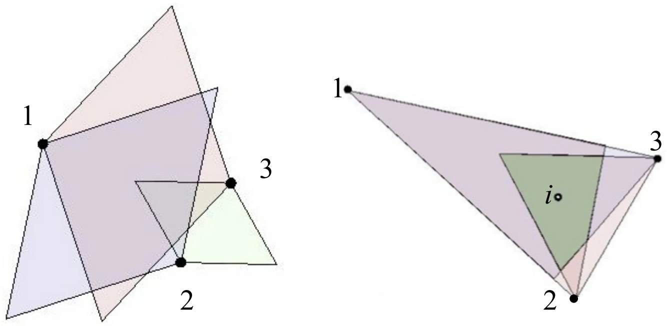


Figure 3.4: The overlapping zones of links 12, 13, and 23 (**left**); the overlapping zones of the links in the same network constrained to the convex hull of the fixed nodes 1, 2, and 3 (**right**). Relay node i is positioned in the overlap of zones $\theta_{12}, \theta_{13}, \theta_{23}$ and can reduce the communication cost of all three links in the network, iff their traffic is routed through i.

cation cost of all the links 12, 13, and 23 if and only if i is placed in the overlap of the set of zones $\theta_{\{12,13,23\}} \in X(A)$.

Observation 3.4. *Consider the set $Y^* \in \mathbf{Y}$ of routing paths connecting all $\langle sd \rangle$ pairs in the relay network G'. Let A be a subset of the utilized links in Y^* from the initial network G. For a given relay node i, from Observations* (3.2) *and* (3.3)*, $\forall u, v$ such that $uv \in A$, if $ui \in Y^*$ and $iv \in Y^*$ then $d_{ui} < d_{uv}$ and $d_{iv} < d_{uv}$. The latter is equivalent to placing i in θ_A. By contraposition, if i is not placed in θ_A, $\exists$ u and v such that $uv \in A, ui \notin Y^*$ and $iv \notin Y^*$. Furthermore, there $\exists$ a relay node i that if placed in θ_A, then the set of routing paths Y^* minimizing network communication cost and utilizing i should include links ui and iv, $\forall u, v$ such that $uv \notin A$.*

From *Observation* (3.4), for instance, if the zones of links u_1v_1 and u_2v_2 do not overlap, then the links u_1i*,* iv_1*,* u_2i*, and* iv_2 cannot all be in Y^*. Each set of routing paths containing all four of these links is suboptimal.

3.2.3.2 Feasible Polygons and Their Independence

Based on *Observations* (3.1) - (3.4), in this section we describe a set of feasible polygons that necessarily contains the optimal positions for the placement of relay nodes in the network. We show that just obtaining this set of feasible polygons is in itself an NP-hard problem.

- *Feasible polygons*:

Let E be the set of links in the initial graph G. Each $\theta_S \in X(E)$, $S \subseteq E$, defines an overlap polygon. From *Observation* (3.4), if relay node i is placed outside θ_S it would be suboptimal for i to offload all links in S. θ_S provides an initial feasible site at which relay node i can be placed to offload all links in S. The relay nodes are optimally placed in a subset of the polygons contained in $X(E)$. E.g., in fig. 3.4 (**right**) the overlap polygon $\theta_S \in X(E)$ associated with the set of links $S = \{12, 13, 23\}$ is in green and contains relay node i. In this example, the routing Y^* in the relay network is not hard to compute. Given i's placement in θ_S, and the traffic routed through i, i's optimal position is found by solving the RPFT with input $\mathbf{Q}_{\mathbf{Y}^*}$.

Definition 3.2. Let $A \subseteq E$ be a set of links connecting some subset of fixed nodes in G. The *set of feasible polygons* is the set $X(A) = \{\theta_S : S \subseteq A, \theta_S > 0\}$.

- *Finding the set of Feasible Polygons is NP-Hard*:

Notice that some zones of links in E may overlap while others may not have common overlap. For instance, Figure 3.5 shows an example of a network with two sets of links S_1 and S_2, each containing three links. There we have $\theta_{S_1} \in X(E)$ and $\theta_{S_2} \notin X(E)$, since $\theta_{S_1} > 0$ while $\theta_{S_2} = 0$. Obtaining $X(E)$ is a prerequisite

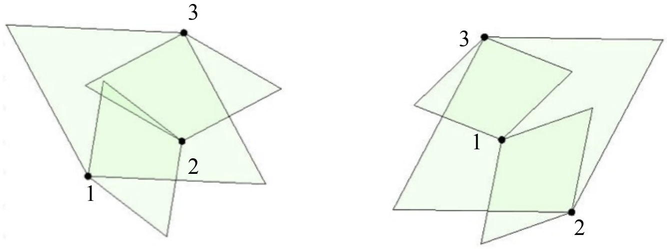


Figure 3.5: $S_1 = S_2 = \{\theta_{12}, \theta_{13}, \theta_{23}\}$. $\theta_{S_1} \in X(E)$ for the graph G (**left**); $\theta_{S_2} \notin X(E)$ for the graph G (**right**). Zones θ_{12}, θ_{13} have positive overlap for the graph on the **left**, and zones θ_{12}, θ_{13} have zero overlap for the graph on the **right**.

for finding optimal placement for the relay nodes. Can we design an efficient algorithm generating the set of overlap polygons that belong to $X(E)$?

Theorem 3.2. *Finding all sets in $X(E)$ is NP-hard.*

Proof. We start by defining the zone graph $G_Z(V_Z, E_Z)$ of $G(V, E)$. Associate a vertex $v_{ij} \in V_Z$ to each zone $\theta_{ij}, ij \in E$. Any two vertices $v_{ij}, v_{i'j'} \in V_Z$ are connected by an edge $e \in E_Z$ if and only if $\theta_{ij} \diamond \theta_{i'j'}$. The zone graphs of three sample graphs on fixed nodes are given in fig. 3.6. Consider the clique complex $\chi(G_Z)$ of the graph G_Z and let $V_S \subseteq V_Z$ be the set of nodes associated with the links in $S \subseteq E$.

We next show that $V_S \in \chi(G_Z)$ if and only if $\theta_S \in X(E), \forall V_S \subseteq V_Z$.

Let $\theta_S \in X(E)$. We have $\theta_{ij} \diamond \theta_{i'j'}, \forall ij, i'j' \in S$. Then, v_{ij} and $v_{i'j'}$ are connected by an edge $e \in E_Z$, $\forall v_{ij}, v_{i'j'} \in V_S$. Therefore, V_S is a clique in G_Z, and $V_S \in \chi(G_Z)$. Conversely, let $V_S \in \chi(G_Z)$, then the vertices in V_S form a clique in G_Z. We have that v_{ij}, and $v_{i'j'}$ are connected by an edge $e \in E_Z, \forall v_{ij}, v_{i'j'} \in V_S$. By the definition of $G_Z, \theta_{ij} \diamond \theta_{i'j'}, \forall ij, i'j' \in S$. This implies $\theta_S \in X(E)$.

There is equivalence between the elements in $X(E)$ and the elements in $\chi(G_Z)$. Listing all the sets in $X(E)$ is equivalent to listing all cliques of G_Z. Listing all

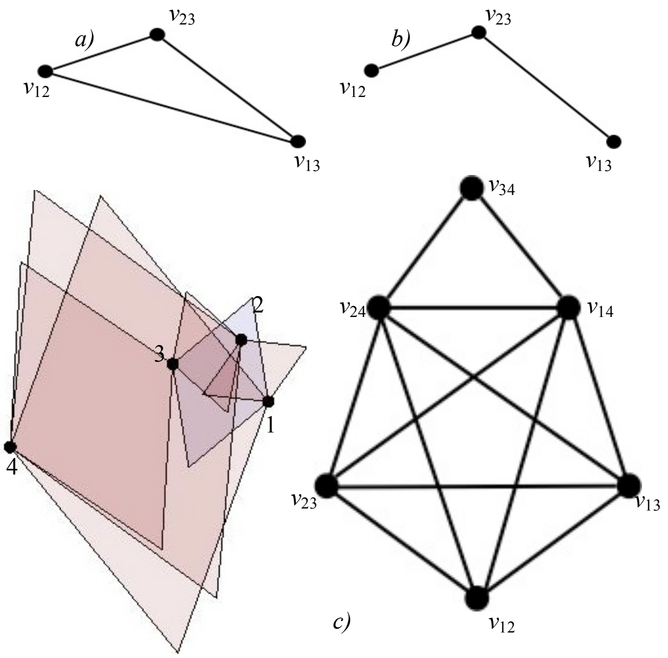


Figure 3.6: Zone graphs of three network instances: a) zone graph of the network in fig. 3.5 (**left**); b) zone graph of the network in fig. 3.5 (**right**); c) network on 4 nodes (**left**) and its zone graph (**right**). The zone graphs quickly become more complex as the number of nodes in the corresponding networks increases.

cliques in G_Z is at least as hard as listing all maximal cliques of G_Z. For general graphs, listing all maximal cliques is an NP-hard problem ([83]). □

Hence, even listing the potential polygons where relay nodes should be placed is NP-hard. Furthermore, for general graphs this clique problem is hard to approximate in polynomial time ([123]).

- Feasible Polygons as an Independence System:

Corollary 3.3. *$X(E)$ is an independence system, therefore $X(E)$ is the intersection of m matroids, where m is finite.*

Proof. From Theorem (3.2), $X(E)$ is equivalent to the clique complex $\chi(G_Z)$ of the graph G_Z. The clique complex of a graph is an independence system ([124]). Therefore $X(E)$ is an independence system. It is well known that any independence system is an intersection of a finite number of matroids ([125]). □

Next, we pose the GRPP as a set function maximization problem, subject to independence constraints. This perspective provides intuition for heuristics solving approximately the GRPP.

3.2.3.3 Optimal Assignment: A Maximization Perspective

Let the communication cost reduction ϕ_K be the difference between the communication cost in the initial network $G(N, E)$ and the communication cost in the relay network $G'(N \cup K, E')$:

$$\phi_K = \sum_{ij \in E} f_{ij} - \sum_{ij \in E'} f_{ij} \tag{3.18}$$

Given $X(E)$, the GRPP problem becomes an optimal assignment problem of elements/polygons in $X(E)$ to relay nodes. Each element in $X(E)$ may be assigned to 0, 1 or more relay nodes. Rename the set of overlaps $X(E)$ to J. Let the pair $(i, j), i \in K$ and $j \in J$, denote the assignment of relay node i to the overlap polygon j. Let $\mathbf{\Phi} = \{(i, j) : i \in K, j \in J\}$ be the set of all possible assignment pairs, and $\mathbf{\Phi}_j = \{(i, j) : i \in K\}, j \in J$. The maximization problem below is equivalent to the GRPP:

$$O^* = \arg\max_{O \subseteq \mathbf{\Phi}} \left\{ \phi(O) : |O \cap \mathbf{\Phi}_j| \leq k, j \in J \right\} \tag{3.19}$$

Notice that since we only renamed $X(E)$ to J in (3.19), $\mathbf{\Phi}$ in (3.19) is an independence system from *Corollary* (3.3). The function $\phi(O)$ is defined here as the maximum communication cost reduction in the network G achieved by placing a subset of the K relay nodes within a combination of polygons in the set $X(E)$, according to the assignment pairs $(i, j) \in O \subseteq \mathbf{\Phi}$. The function $\phi : O \to \mathbb{R}^{+}$ is not given explicitly. However, for a particular $O \subseteq \mathbf{\Phi}$ we can compute $\phi(O)$.

We next show how to evaluate the minimum communication cost corresponding to the set of assignment pairs in O. This is equivalent to evaluating $\phi(O)$. Then, we describe the **BruteForceMax** algorithm solving (3.19).

We start by noting from (3.18) that maximizing the communication cost reduction is equivalent to minimizing the communication cost in G'. Given each $(i, j) \in O$ for a particular set O, we position a relay node i within the overlap $j \in J \Leftrightarrow \theta_S \in X(E)$ at a random point in θ_S. Suppose the traffic matrix $\mathbf{Q}$ of the initial network G is given. This can be obtained by running Floyd-Warshall's algorithm on the weighted adjacency matrix C of G. The communication cost in G, given the resulting routing, is computed using (3.9). We apply *Observation* (3.4) and offload the traffic from all links in S only through the relay nodes placed in θ_S. Given this constraint, let Y_O be a set of routing paths on the vertices $N \cup K$, while the relay nodes are placed according to the assignment pairs in O. Let $\mathbf{Y_O}$ be the collection of all possible sets Y_O for a fixed O. Run the **BruteForceMin** algorithm with input $\mathbf{Y} = \mathbf{Y_O}$.

This yields the minimum communication cost $F^{\mathbf{Q_Y}}(\mathbf{v})$ in the network G' for the fixed O. Equivalently, we have $\phi(O)$.

The **BruteForceMax** algorithm follows:

1. For each $O \subseteq \mathbf{\Phi}$, compute $\phi(O)$ as described above.
2. Pick O^{*} that maximizes $\phi(O)$ over all $O \subseteq \mathbf{\Phi}$.

Similarly to **BruteForceMin**, the **BruteForceMax** algorithm is exponential on the number of vertices in $N \cup K$. The formulation of the GRPP as maximization problem does not in itself reduce the complexity of the solution, and we do not know the properties of the function $\phi(O)$ from (3.19). However, we know that (3.19) is a maximization problem over an independence system $\mathbf{\Phi}$. Typically, greedy algorithms perform well in that context. The **RePlace** heuristic algorithm presented next follows such Greedy strategy.

3.2.4 The *RePlace* Heuristic

Given the exponential running time of the discussed brute force algorithms and the hardness of the exact computation of the feasible polygons in $X(E)$, in this section, we resort to designing an efficient heuristic for the GRPP.

Based on the initial network $G(N, E)$ and demand matrix $\mathbf{W}$, we start by computing the adjacency cost matrix $\mathbf{C}$ of G using (3.6). We find routing Y and traffic matrix $\mathbf{Q}$ by running, for instance, all pairs shortest paths routing algorithm over the weighted adjacency matrix $\mathbf{C}$ of G. Suppose we can enumerate all sets $\theta_S \in X(E)$. In the next section, we describe a practical implementation that computes an overlap polygon θ_S from the overlapping zones of the links in S. We also give a procedure to approximately compute the set $X(E)$ of overlap polygons.

Given input G, $\mathbf{C}$, Y, $\mathbf{W}$, $\mathbf{Q}$, $X(E)$ and K, the goal is to find k positions at which to place the relay nodes and the routing through the resulting relay network, so that a best-effort solution to (3.19) is computed. The **RePlace** algorithm given by **Algorithm** (3.1) relies on a Greedy heuristic to achieve that.

Suppose P is the set of placed relay nodes after iteration t. v_t contains the coordinates of the nodes in P. Correspondingly, $G'(N \cup P, E_t)$ is the resulting network with weighted adjacency matrix $C_t = [c_{ij}]_{(n+t)(n+t)}$, where $E_0 = E$. Let F_t be the communication cost in the network after iteration t. Then, ϕ_t is the communication cost reduction:

$$\phi_t = F_t - F_0 = \sum_{ij \in E_t} f_{ij} - \sum_{ij \in E_0} f_{ij} \tag{3.20}$$

At each iteration t of the **RePlace** algorithm, we place a new relay node p at a location within the overlap polygon $\theta_S \in X(E)$ that leads to the maximum ϕ_t (lines 10 - 38, **Algorithm** (3.1)). To determine θ_S maximizing ϕ_t at each iteration t, **RePlace** probes all overlap polygons prior to placing each relay node p.

Namely, at iteration t, for each $\theta_S \in X(E)$, relay node p is temporarily placed at a random location in θ_S (line 11, **Algorithm** (3.1)). The network routing is updated according to *Observation* (3.4), to account for the placement of p (lines 13 - 21, **Algorithm** (3.1)). Next, node p's position within θ_S is optimized by solving (RPFT), with respect to the updated network routing (lines 22 - 28, **Algorithm** (3.1)). Notice that the positions of all relay nodes placed until iteration t are also optimized, within the constraints of their respective overlap polygons. Then, the estimated communication cost reduction ϕ_S from placing p in overlap θ_S is obtained (line 28, **Algorithm** (3.1)). Eventually, p is placed in overlap θ_S leading to maximum ϕ_t at iteration t.

RePlace terminates when all k relay nodes are placed and returns the Greedy solution $\mathbf{v_g}$ for the positions of the relay nodes, along with the corresponding routing Y_g in the network.

Algorithm 3.1: RePlace

```
input: $G, C, Y, \mathbf{W}, \mathbf{Q}, X(E)$, and $K$
output: $\mathbf{v_g}, Y_g$: vector of relay node locations and a set of routing paths
$t \leftarrow 0$; $P \leftarrow \emptyset$; $Y_t \leftarrow Y$; $\mathbf{v}_t \leftarrow \emptyset$; $C_t \leftarrow C$; $E_t \leftarrow E$
while $|P| \leq |K|$ do
    $p \leftarrow rand(K \setminus P)$ // pick a relay node from the set of not placed relay nodes
    $P \leftarrow P \cup p$
    $E^S \leftarrow E_t$
    $t \leftarrow t + 1$
    $\phi_{max} \leftarrow 0$
    $\mathbf{v_t} \leftarrow v_{t-1}$
    $Y_t \leftarrow Y_{t-1}$
    for all $\theta_S \in X(E)$ do
        $(x_p, y_p) \leftarrow (x_{rand}, y_{rand})$ // tentatively place $p$ at a random location in $\theta_S$
        // estimate communication cost reduction of placing $p$ in $\theta_S$
        for all $v \in N \cup P$ do
            if $d_{vp} \leq R$ then
                $c_{vp} = \frac{1}{1-r_{vp}}$ // set the $c_{vp}$ entry in $C_t$
                $E^S \leftarrow E^S \cup vp$
            end if
        end for
        $C^S \leftarrow C_t$
        $Y^S \leftarrow$ FloydWarshall$(G(N \cup P, E^S), C^S)$ // compute all-pairs shortest paths $Y^S$ using Floyd-Warshall on $G(N \cup P, E^S)$ with matrix $C^S$
        $Y \leftarrow Y^S$
        for all $ij \in E^S$ do
            // compute traffic $q_{ij}$ on link $ij$ per (3.7); set entry $q_{ij} \in \mathbf{Q}_Y$
            $q_{ij} \leftarrow \sum_{\langle sd \rangle \in Y^{ij}} w_{sd}$
        end for
        $v_S \leftarrow$ ConstrainedDescent$(F^{\mathbf{Q}_Y}(v_t \cup (x_p, y_p)))$ // Solve the RPFT problem (3.12) with input $\mathbf{Q}_Y$ via constrained gradient descent
        $F^S \leftarrow F^{\mathbf{Q}_Y}(\mathbf{v}^S)$
        $\phi^S \leftarrow F_0 - F^S$
        if $\phi_{max} \leq \phi^S$ then
            $\phi_{max} \leftarrow \phi^S$
            $\mathbf{v}_t \leftarrow \mathbf{v}^S$
            $Y_t \leftarrow Y^S$
            $E_t \leftarrow E^S$
        end if
        $E^S \leftarrow E_{t-1}$
        $C_t \leftarrow C_{t-1}$
        $\phi_t \leftarrow \phi_{max}$
    end for
end while
$\mathbf{v_g} \leftarrow \mathbf{v}_t$
$Y_g \leftarrow Y_t$
```

3.2.4.1 RePlace Implementation

The implementation of the above **RePlace** heuristic assumes that one can compute an overlap polygon θ_S from the overlapping zones of the links in a set $S \subseteq E$, and also that one can find all sets $\theta_S \in X(E)$.

Given graph G, the collection $X(E)$ is approximately found via Monte Carlo based approach. Suppose M points are sampled on the plane uniformly at random. It is very efficient to check whether each of the M points is positioned within any given link's zone. Let m be a sample point and S_m be a set of zones that contain m. Combining the sets S_m for each m yields approximate set $X'(E)$. The larger M the more complete the approximate solution $X'(E)$ of $X(E)$.

We can efficiently find the overlap polygons of the zones forming $\theta_S \in X'(E)$. To do that, we adapt the clipping algorithm of ([126]). Given a set of convex zones as input, we use clipping to compute the intersection polygons of the zones in $X'(E)$.

3.2.5 Numerical and Simulation Results

This section studies the performance of the **RePlace** heuristic described above, which is compared numerically with the performance of **RePlace** to the performance of an optimal brute force algorithm. (In small networks, one can find the optimal solution to the GRPP via either of the brute force algorithms described above.) The performance of the **RePlace** heuristic matches the performance of the optimal solutions very closely, and in most of the trials the two are identical. Using the JiST/SWANS ([119]) full network stack simulator, we evaluate the performance of **RePlace** in terms of network delay and communication cost reduction. We show that under practical wireless network conditions where interference, collisions, links asymmetries, fading, etc. are present, the optimized placement of small number of relay nodes reduces the delay by a factor of 2 and almost completely eliminates overhead communication cost in the resulting network as the number of relay nodes increases.

3.2.5.1 Numerical Evaluation

The performance of **RePlace** is shown in fig. 3.7. In small networks, we directly compare **RePlace** to the optimal solution obtained by the **BruteForceMax** algorithm. Notice that the two algorithms perform remarkably close. Each data point represents the communication cost reduction achieved by placing k relay nodes in a network of n nodes and is averaged over 50 random network topologies. $X(E)$ is computed exactly by brute force for the input of the **BruteForceMax** algorithm. We compare the results for the optimal and **RePlace** solutions for networks of size up to $n = 10$ and $k = 6$ nodes.

Figure 3.8 lists a few typical relay network topologies found by the two algorithms for different values of n and k. In these examples, **RePlace** matches exactly the optimal solution in terms both of routing and relay nodes' positions.

3.2.5.2 JiST/SWANS Simulations

To investigate the effect of bandwidth limitation, interference, collisions, link asymmetries, etc., we simulate **RePlace** in the full stack network simulator

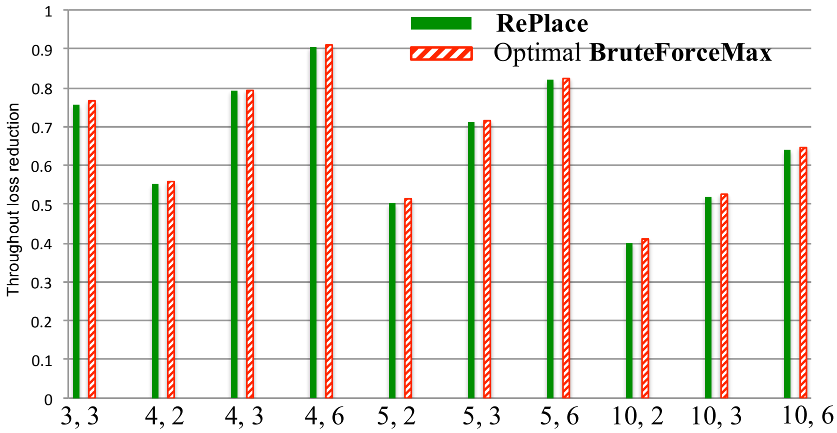


Figure 3.7: **RePlace** and the optimal **BruteForceMax** communication cost reduction performance in small networks for varying numbers n and k of fixed and relay nodes ($x - axis$: n, k). The **RePlace** heuristic achieves solutions remarkably close to the optimal ones.

Table 3.1: Parameters of the JiST/SWANS Simulation

Simulator	JiST/SWANS v1.0.6
Radio frequency	2.4GHz
Channel bandwidth	1Mb/s
P_t	5[dBm]
R	$\approx$ 10[m]
Area	Varying
Propagation model	Free Space
MAC Layer	IEEE 802.11b
b	256[bits]
n	[5,50]
k	[2,30]

JiST/SWANS. Table (3.1) summarizes the simulation setup parameters. Each simulation run consists of two experiments. First, only the nodes in N are deployed in the network. As above, we model the link costs c_{ij} according to the log-distance path loss model and then find the lowest-costs paths routing on the

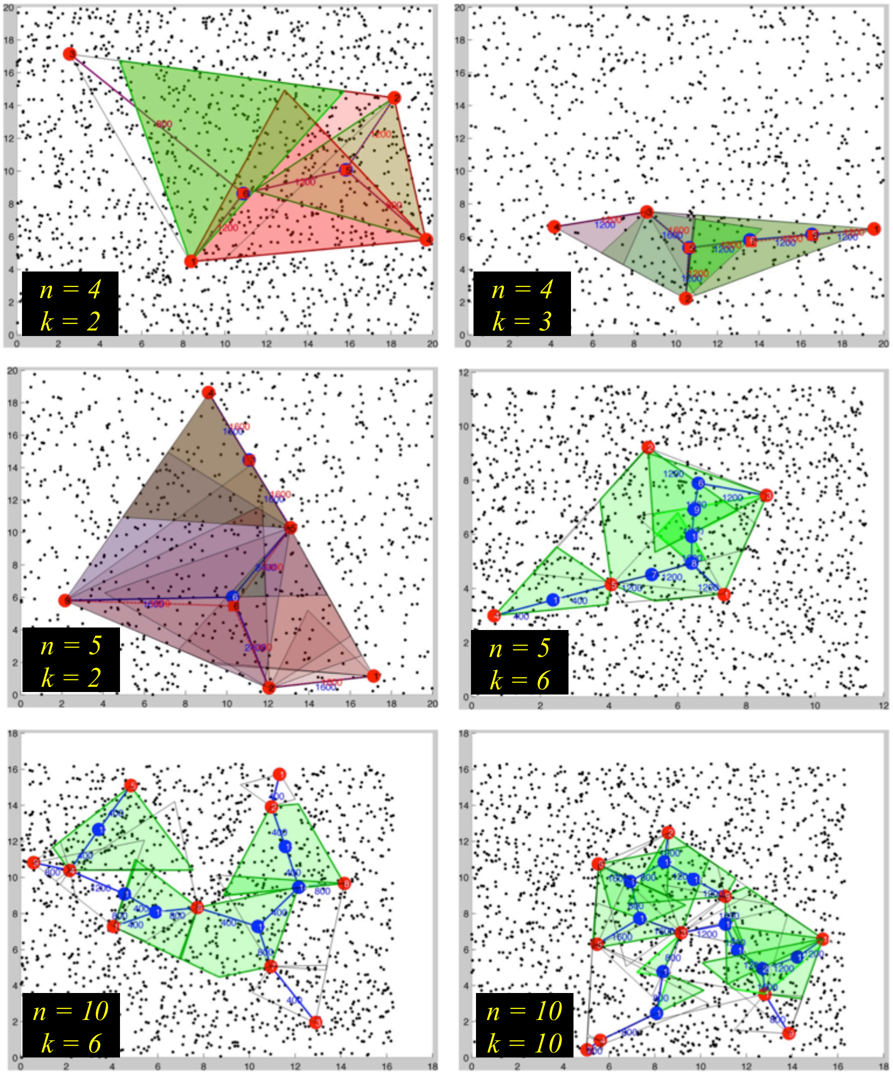


Figure 3.8: The output of the **RePlace** and the optimal **BruteForceMax** algorithms on typical random network topologies. The relay nodes placed by the **RePlace** algorithm are marked with blue circles; the relay nodes placed by the **BruteForceMax** algorithm are marked with red squares. In these cases the solutions, both routing and positions of nodes, output by the two algorithms are the same and the respective relay nodes placements overlap. Hence, only one solution is visible at a time. For $k = 2$ and 3, all zone overlaps bounded by the convex hull of the network are shown. For $k = 6$ and 10, only the zone overlaps selected for the optimal placement of the relay nodes are visible in green. The black dots are the sample points used for computing the approximate set $X(E)$ input to the **RePlace** algorithm.

n nodes. The routing is static and centralized. Each node receiving a packet with a given source and destination has access to the routing table and determines the next hop of the packet. In the second experiment, the nodes in $N \cup K$ are deployed. The coordinates of the n nodes and $\mathbf{W}$ are provided as input to **RePlace**. We model the traffic demand q_{ij} on each link ij per (3.9). Using **RePlace**, we obtain the positions of the k relay nodes and the corresponding routing over the nodes in $N \cup K$.

For both experiments, we consider the metrics communication cost reduction and average packet delay (i.e. the delay of a packet successfully delivered at a destination, averaged across all such packets on all network paths). These metrics are investigated in networks of different sizes and varying number of relay nodes. Each data point in the simulation figures represents the mean network performance over a 100 different random network instances.

- *Communication Cost Reduction*:

The communication cost reduction obtained by **RePlace** (vis-á-vis the base case where no relay nodes are deployed) is shown in fig. 3.9. The communication cost metric we consider accounts for the number of dropped packets due to low SINR at each receiver in the network. More specifically, in IEEE 802.11 the transmitter does not receive an acknowledgement if a packet is dropped at the receiver. This causes the transmitter to retry sending the packet. The number of such retries/retransmissions effectively leads to loss in the throughput of the network. The communication cost reduction metric measures the fractional difference between the number of retransmissions when relay nodes are placed and the number of retransmissions without placement of relay nodes. Note that the low SINR at the receiver can result from the combination of two phenom-

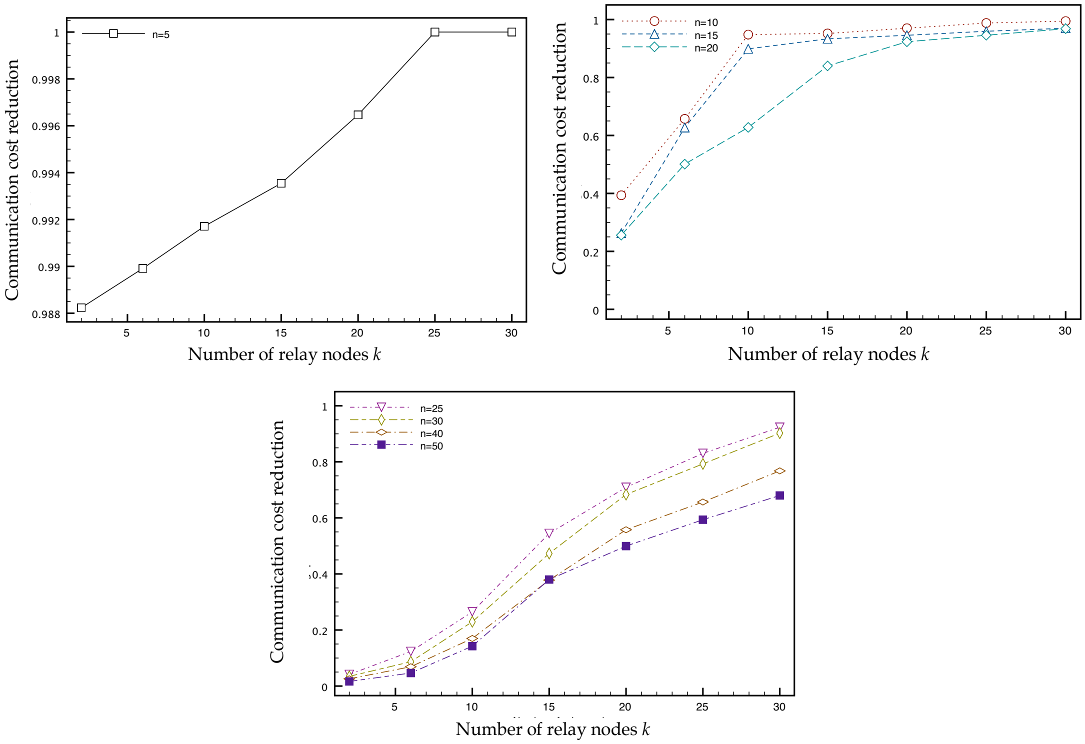


Figure 3.9: The fractional difference between the number of retransmissions due to dropped packets when relay nodes' positions are optimized in comparison to the base case with no relay nodes placed. As the ratio k/n varies there are three different trends: **top-left**, **top-right** and **bottom**. As expected for large k/n, the reduction is almost 100%. The communication cost reduction rate is highest for smaller values of k/n. The communication cost reduction rate becomes lower for larger k/n.

ena: interference from other transmitters on one hand and low received signal power on the other. It is observed, similarly to the work in [59], that in smaller networks operating in mid-SNR regime, interference does not affect critically the packet loss in the network. As long as the relay nodes are placed optimally so that signal degradation due to separation distance is minimized, packet loss can be reduced significantly. For instance, in a network of 20 nodes we can position 6 relay nodes and achieve almost 70% reduction of dropped packets as shown in fig. 3.9.

- *Average Packet Delay*:

The communication cost reduction leads to substantial decrease of average packet delay in the network when relay nodes are optimally deployed. This is due to the retransmission backoff mechanism in the IEEE 802.11 MAC protocol. Each retransmission, until an acknowledgement is received by the receiver, leads to increasingly larger delay. Reducing the number of retransmissions effectively reduces the packet delay. The delay ratio $DR_{k,n}$ is defined as **average end-to-end packet delay in a network of *n* fixed nodes with *k* relay nodes** *divided by* **the average packet delay in the network of n fixed nodes without relay nodes**. If the ratio is less than 1, the delay is lower after the addition of the relays.

The plots of $DR_{k,n}$ for the different values of k and n are shown in fig. 3.10. Notice that for all values of n and k the achieved delay when relays are deployed utilizing **RePlace** is less than the delay when no relays are deployed.

Figure 3.10 (**top**) depicts the delay ratio for large ratio k/n. In these cases the communication cost reduction saturates, the average path length grows longer and the delay increases as more relay nodes are added to the network. Furthermore, the delay increases due to the RTS/CTS mechanism on the MAC layer as the number of relay nodes in the network grows. A similar effect is observed in fig. 3.10 (**middle**) for $k/n > 1$: the delay increases as more relay nodes are added in these cases. In contrast, fig. 3.10 (**bottom**) shows the *decrease* of delay, as more relay nodes are added while $k/n < 1$. In terms of delay, the optimum ratio k/n is approximately 1.

- *Performance Comparison*:

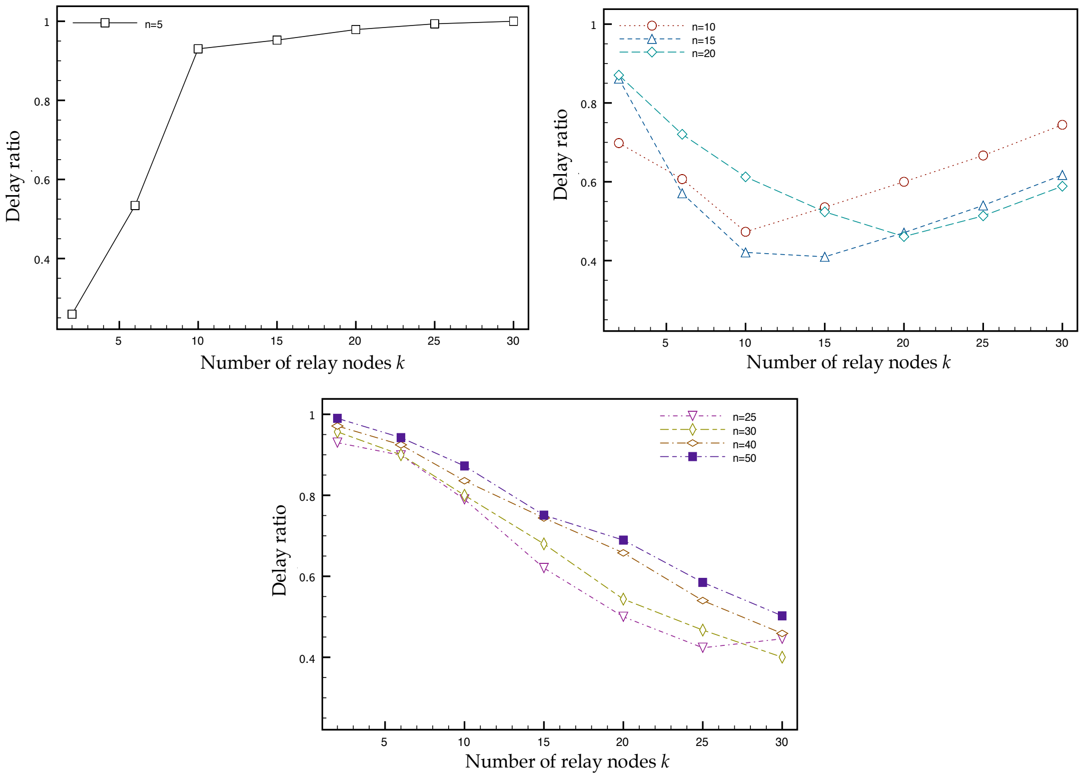


Figure 3.10: Delay ratio $DR_{k,n}$. The delay, after optimized placement of the relay nodes, is less compared to the base case with no relay nodes, for the different values of n and k shown. As the communication cost reduction saturates (for large k/n as shown in fig. 3.9), the delay ratio increases accordingly. In these cases increasing the number of relay nodes causes the average routing path length in the network to increase, without substantial communication cost reduction. The delay ratio is also high and closer to 1 for small k/n: too few relay nodes do not reduce the communication cost substantially.

We investigate the performance of two strategies alternative to **RePlace**, for placing relay nodes in wireless networks. The **Steiner tree**-based strategy follows the General Steiner Tree (GST) model discussed in section 3.1 above. GST is utilized in [58] and [59], among other works. The **DoubleStage** strategy has been recently suggested in [55]. The authors of [55] account for the energy cost required for reconfiguring the positions of the relay nodes. Since in the model considered here, there is no such reconfiguration cost, it is assumed that the reconfiguration cost is set to 0, when the relay placement algorithm of [55] is utilized. **DoubleStage** constructs a routing tree in the original graph (sans relay nodes) by finding a shortest path connecting a source-destination pair. The length of each edge in the path models the energy required for successful transmission of a packet on the corresponding wireless link. Hence, the resulting shortest path minimizes the communication cost. This is the first stage of the algorithm. In the second stage, the authors position k relay nodes iteratively. Each relay node's position is determined, so that the communication cost in the network is minimized given the routing found in the first stage of the algorithm. The routing is not updated during the placement of the relay nodes. This inherently leads to potential inefficiencies of the **DoubleStage** algorithm's output.

The performance of the **Steiner tree** and **DoubleStage** algorithms relative to the **RePlace** algorithm is shown in fig. 3.11 (**left**: communication cost; and **right**: delay ratio). The number k of placed relay nodes is the same for each of the three schemes. **RePlace** outperforms the other two schemes for varying numbers of fixed nodes in the network.

3.2.6 Approximating Special Cases

The maximization problem formulation of the general relay placement problem given in (3.19) generalizes a number of cases, in which a constant approximation algorithms to the optimal solutions are known. Some of them may have practical significance. Below, a brief sample of these results are provided. The different cases result from placing different constraints on the communication

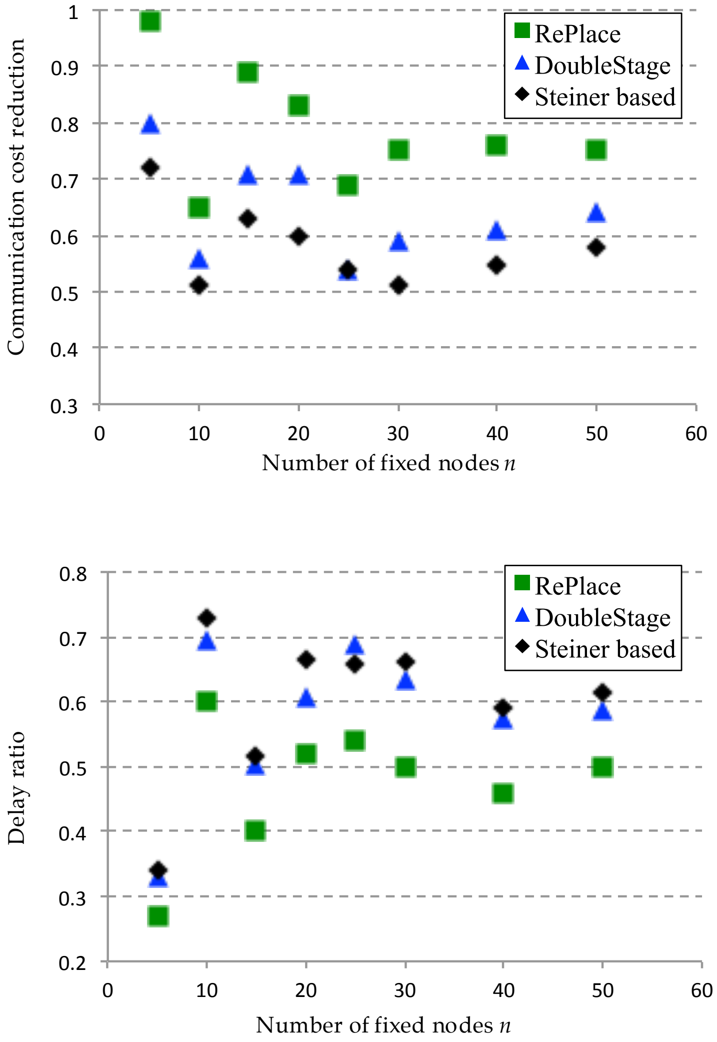


Figure 3.11: **Top**: *Communication cost* reduction achieved by **RePlace** vs. **DoubleStage** vs. **Steiner-tree** based approaches. Higher values of communication cost reduction are better. **Bottom**: *Delay ratio* of **Re-Place** vs. **DoubleStage** vs. **Steiner-tree** based approaches. Lower values of the delay ratio are better. The three schemes are given equal budgets *k* of relay nodes.

cost reduction function $\phi(O)$ defined in (3.19). Throughout, it is assumed that $X(E)$ is given by an oracle.

- *Case 1*: *$\phi(O)$ is nonmonotone and submodular function over the independence system* $\mathbf{\Phi}$. There is a Greedy algorithm that iteratively selects assignment pairs $(i, j) \in \mathbf{\Phi}$ and provably achieves a constant factor $((m-1)/m^2 + \epsilon)$-approximation, $\forall \epsilon > 0$ ([127]). Here m is the number of matroids intersecting to form the independence system $\mathbf{\Phi}$.

Potential application problem is the location of network integration points in wireless networks. This may be viewed as a special case of the GRP where relay/integrator nodes do not communicate with each other but only communicate with the fixed nodes. Also, each fixed node communicates with exactly one integrator/relay node. The case of placing a single relay node in the network also falls in this category. The online problem of placing relay nodes given one at a time (where the positions of the previous relay nodes cannot be altered) falls in this category as well, given that the routing through the already placed relay nodes is not affected by the addition of new relay nodes.

- *Case 2*: *$\phi(O)$ is monotone and submodular function over the independence system* $\mathbf{\Phi}$. Similarly to the above case there is a constant approximation Greedy algorithm. The constant factor is $1/(m+1)$ ([125]). A potential application problem would be the integrators placement problem, where communication cost on the network links between integrators and fixed nodes is relatively uniform.

- *Case 3*: *$\phi(O)$ is nonmonotone and submodular function and the independence system $\mathbf{\Phi}$ is an intersection of a single matroid (i.e. $\mathbf{\Phi}$ is a matroid).* In this case, there is a $1/3.23$ constant approximation algorithm ([128]). Of course, this case is rather restrictive and may model only very specific practical scenarios.

- *Case 4*: *$\phi(O)$ is monotone and submodular function and the independence system $\mathbf{\Phi}$ is a matroid*. There is a well known 1/2-approximation algorithm in this case (e.g. [125]). This is the most restrictive of all cases in terms of modeling potential practical scenarios, and is included here only for completion.

3.3 Other Relay Placement Schemes and Goals

The problem of placing relay nodes in a network with the goal of improving networks' links reliability and reducing communication costs is an important one, with many applications. Hence, a number of works in the technical literature tackle different versions of it, in different contexts.

For instance, a number of works study the placement of relay nodes to provide connectivity in the network (e.g. [129], [130],[131],[43] and references therein). The problem addressed by the authors in this setting is different from the problem considered above. Given a network of fixed nodes, the goal is to place a minimum number of relay nodes so that the induced overall network topology is connected. In some cases (e.g. [130]) to accomplish that, the authors assume that the relay nodes have larger transmission radius than the fixed nodes; in other cases (e.g. [129]) a constant transmission radius is given as an input to the problem, and the number of required rely nodes is found so that the network is connected. The authors of [131] consider a heterogeneous network where a minimum number of relay nodes are utilized to connect sensor nodes to more powerful base stations via directed paths. In [43], the authors extend this result allowing constraints to be placed on the discrete sets of locations available for relay node placement (e.g. due to natural obstacles). They required the construction of bi-directional paths comprising relay nodes and connecting the base stations and sensor nodes. Although the network lifetime may be increased as a result of the improved network connectivity, the above works approach the use of relay nodes in a different setting than the one presented in this chapter. Here the relay network topology is explicitly optimized with the goal of reducing communication costs. Relays' communication radius is not altered and the physical communication model from [117] is assumed rather than the protocol model assumed by the above studies. Furthermore, the solution of the connectivity problem relies on approximate optimal coverage algorithms, often based on *steinerization* (e.g. [43]) techniques. As observed in section 3.1, the latter models do not reflect well the characteristics of the relay placement problem discussed in this chapter.

A different stream of technical literature (e.g. [45], [46], and [132]) considers the deployment of relay nodes to provide fault tolerance in networks guaranteeing *m*-connectivity between sensor nodes, or sensor nodes and base stations. In a comprehensive work, Patel et al. ([133]) study versions of these problems restricted to a setting where sensors, relay nodes and base stations may occupy only a discrete set of locations on a 2D grid of points. Minimizing congestion ([134]) and load balancing are other application areas where topology control in the form of relay node placement is utilized (e.g. [46]). However, network performance in terms of communication cost is not an objective there.

Targeting problems more similar to the one considered in this chapter are studies in the area of controlled mobility, where nodes in the network may adjust their locations to optimize certain network performance metrics (e.g.[135],

[118], [55]). In these settings relay nodes may move on a continuum of points for optimal performance. For instance, the authors of [135] allow a set of k mobile nodes to adjust their location and provide a wireless communication backbone (i.e. comprising relay nodes) that minimizes energy expenditure by increasing the reliability of links. The authors propose a solution finding the lowest energy level of a dynamical physical system modeling the problem. However, in their setting routing is not considered. Similarly, in [118] the authors construct a wireless backbone communication network of relay nodes maximizing links' quality using Steiner tree approximations. They study the performance of their schemes with different routing patterns, but the location of the relay nodes is independent of the routing and traffic loaded on the links. More recently, the authors of [55] employ mobile relays to minimize both energy due to transmissions and the movement of the relay nodes. Their algorithm assumes that the routing tree does not change during nodes' mobility. This is analogous to the RPFT problem defined in section 3.2.2.2 above, where the assumption is that the routing pattern is fixed. In this case, the optimal positions of the relay nodes is obtained by convex optimization. In contrast, the algorithm presented in [55] does not guarantee an optimal solution in this case.

Finally, a number of works on sensor networks have considered a form of sensor nodes placement in order to optimize network communication cost. For instance, in [59], the authors utilize Steiner tree model to place relay nodes in a sparse network and increase links' reliability. They observe that in the latter setting the log-normal path-loss model is rather accurate and interference does not contribute significantly to reduce the PRR. This conclusion is also corroborated by the simulations presented here in section 3.2.5.2. The authors of [58] consider the placement of relay nodes, so that overall link cost is minimized while the gathered information by sensor nodes is maximized. Their algorithm approximates a General Steiner Tree to suggest locations for the communication relay nodes. In both works, routing in the network and its influence on communication costs and relay nodes' positions is not accounted for.

CHAPTER 4

PROCEDURAL LEVEL: BROADCAST AND RETRANSMISSIONS

4.1 Towards Optimal Broadcast in Wireless Networks

As noted above, broadcast is a fundamental network operation allowing a source node to send a message to all other nodes in the network. In the context of networks where all communications are carried over a wireless medium and network nodes are limited in energy and computational power (e.g. sensor and mesh networks), an efficient broadcast mechanism is exceptionally important for the overall network performance. Similarly, a robust broadcast solution is required in networks where topology can change rapidly due to nodes' mobility (e.g. networks of unmanned aerial vehicles (UAVs), or interconnected wearable devices).

For instance, among the various proposed wireless network routing protocols (e.g., AODV, DSR, OLSR, TRBF, ZRP), a prominent subgroup, referred to as on-demand or reactive routing protocols, is designed based on the philosophy that the discovery of a route in the network should be done only when there is an actual need to route traffic. The route discovery mechanism in on-demand routing protocols relies on some variant of broadcasting to locate a path between the source and the destination nodes. Also, in highly reconfigurable topologies, where the lifetime of network routes may be shorter than the duration of a communication (especially in the case of connection-oriented communication) broadcast, by itself, could be used as a routing mechanism. Yet in other scenarios, data dissemination to all nodes in a sensor network is needed and broadcast is an obvious solution. Being such an essential network operation, it is not surprising the importance of an efficient broadcast implementation has been widely accepted by the networking community.

The remaining of this chapter includes the design of the Time Sequence Scheme (TSS), a new broadcasting algorithm that ranks and orders in time the transmissions of broadcasting nodes, so that the overall number of re-broadcasts in the network is minimized. TSS utilizes only 1-hop topology information while simultaneously achieving full coverage of the network with close to optimal number of broadcast messages and with low delay.

Furthermore, the algorithm is robust to *rapid topological changes* and *network partitions*. It retains its performance in a full network stack implementation, where packet loss at network and MAC layers, for instance, may be present. The various differences and advantages of TSS are discussed and compared with different state-of-the-art approaches to broadcast. It is demonstrated that the described TSS scheme outperforms these approaches in the metrics considered. Also, TSS does not require positional information (unlike a number of recent algorithms) as the latter is infeasible or costly to get in many network settings.

4.1.1 Desiderata for an Efficient Broadcast Algorithm

Assume the network is represented as a connected graph $G = (S, E)$, where S is the set of all the network nodes and E is the set of all the links. Given a source node $s_0 \in S$ that transmits a broadcast message m, suppose it is desirable to reduce the number of rebroadcasts of m required to propagate m in the entire network. Consider the set of nodes $Q \subseteq S$ such that for each node $v \in S/Q$, v has a neighbor in Q. If Q is a connected subgraph of G, Q forms a connected dominating set (CDS). Notice that after the termination of any broadcast scheme that propagates m to all nodes in S, the set of nodes that the scheme has picked for broadcast forms a CDS. For pure flooding ([11]) this is the trivial CDS: S.

A Minimum Connected Dominating Set (MCDS) of G is a CDS in G with minimum cardinality. If just all the nodes in a MCDS broadcast message m, all nodes in G will receive m and the number of broadcast nodes is minimized. In the context of wireless networks, observe that, first, minimizing the number of rebroadcasts (and rebroadcasting nodes) would expend substantially less energy and bandwidth especially as compared to flooding ([136], [137]). Second, recent practical variants of flooding, such as Glossy [138] and Flash [137], can be very rapid and m reaches all nodes with remarkably low latency sans finding MCDS. However, as noted in [137], in many cases the design and application of these flooding schemes is orthogonal to the task of finding MCDS. I.e., the number of transmissions can be minimized by, first, finding the set, Q, of network nodes forming a MCDS; and, second, constraining the flooding only within Q one can minimize latency. Such approach was shown to significantly reduce the energy cost of the Flash flooding protocol in [137], for instance.

Challenges: As noted above, the importance of MCDS to solving and modeling the wireless network broadcast problem has been observed by the research community. However, finding a MCDS of a graph G is an NP-hard problem (even if G is a Unit Disk Graph (UDG) [139]) and one needs to consider an appropriate heuristic algorithm for only approximate CDS. The desired features and challenges for a such practical and efficient algorithm for dynamic wireless reconfigurable networks are that it should

1. reach all the network nodes;
2. transmit the broadcast message as few times as possible (or, equivalently,

reduce the number of times that the broadcast message is received by a network node, optimally to only once);

3. minimize delay (i.e., the time needed for the broad-cast message to be received by the entire network);
4. require only locally available information (e.g., only knowledge of the 1-hop neighborhood topology);
5. minimize the effects, on (1), (2), and (3) above, of topological changes during a broadcast propagation caused by mobility, and due to packet loss.

Unfortunately, few of the so-far-proposed approaches to the broadcast problem satisfy all the above desiderata. Pure flooding protocols violate (2) above; furthermore, they may lead to the notorious "broadcast storm" problem ([11]). Probabilistic schemes ([140], [141], [142], [143], [144], [145], [146], [147]) do not satisfy (1) above. To ensure full coverage, stochastic schemes would need to transmit with probability close to 1, whereby the scheme degrades to flooding. Also, albeit simple, fast, and flexible they are not as efficient in terms of finding small size CDS. Following a different strategy, backbone-based algorithms have also been proposed (e.g., the reader is referred to [148] for comparison of such schemes). They rely on finding approximate MCDS which are constructed by identifying dominating sets, maximal independent sets, or Steiner trees prior to broadcast. While satisfying (1) - (3) above and often guaranteed to find a CDS with size within a constant of the MCDS, such algorithms do not tolerate dynamic topologies well ([43], [149], [150], [151]). Thus, they are incompliant with (5). Furthermore, some of these algorithms are centralized ([152], [153]) and violate (4). Two examples, (e.g. [154], [136]), of broadcast algorithms that satisfy (1) - (5) have been described most recently. In [154] though, the algorithm requires the knowledge of the nodes' geographical position, which is not always feasible. The proposed algorithm in [136] requires only 2-hop local topology information

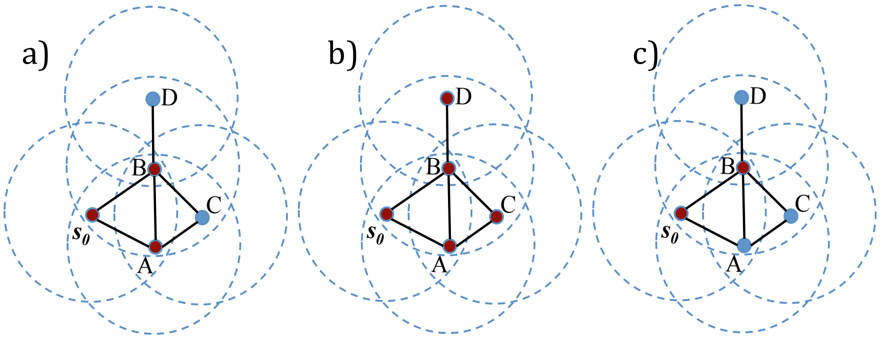


Figure 4.1: The output of four algorithms given a sample network on 5 nodes, where the source node is s_0. The broadcast nodes in the output CDS are in red. a) RBS and CCS output broadcast nodes s_0, A, and B; b) Funke's algorithm outputs s_0, A, B, C, and D; c) TSS outputs the optimum solution which is s_0 and B.

in lieu of geographical position and thus satisfies more of the desiderata for efficient broadcast. However, as the authors of [136] note, in highly mobile networks the algorithm's performance drops due to the 2-hop topology knowledge requirement.

The *Time Sequence Scheme* (TSS) described in the following sections and also in ([155]), addresses these challenges and improves on the state-of-the-art schemes. The TSS's design carefully avoids subtle algorithmic inefficiencies that undermine the performance of broadcast schemes in the current technical literature. Illustrating some of these inefficiencies, fig. 4.1 shows a very simple network topology that is input to three algorithms among the top performing in the context of reducing the number of broadcast transmissions and finding small size CDS.

Figure 4.1 a) shows the output of the *position-aware Responsibility Based Scheme* (RBS), suggested by Khabbazian and Bhargava [154]. Per RBS a network node *i* receiving broadcast message *m* transmits *m* if 1) among *i*'s neighbors there is a node j that has not received m yet and 2) the Euclidean distance between i and j is smaller than the Euclidean distance between j and any other of the neighbors of i that has already received m. Given source node s_0, this broadcast rule requires the transmission of nodes A (since $AC < BC$) and B (since D has not received m after A broadcasts).

Figure 4.1 a) also shows the output of the more recent novel broadcast algorithm utilizing only 2-hop topology information (sans position information) developed by Khabbazian and Bhargava in [136]. This scheme is referred to here as the *Coverage Condition Scheme* (CCS). In a nutshell, each node j maintains a list L_j of neighbors. L_j is gradually pruned by j. Based on information piggybacked in each retransmitted broadcast message and 2-hop neighborhood topology knowledge, j removes nodes from L_j. According to j the removed nodes have received or will receive m from a different node in the network. Per CCS, only nodes selected for broadcast transmit m. If L_j is non-empty, node j is eligible to select node $i \in L_j$ to broadcast next only if j itself has been previously selected for broadcast by a neighboring node k. The selection of i by j is done at random from L_j. If at some point L_j is empty the *coverage condition* of j is satisfied, and j does not need to broadcast if it has not been previously selected to broadcast by k. Otherwise, if j was selected to broadcast by k, j broadcasts but does not select a forwarding node. Here s_0's list contains the nodes A and B. Node s_0 selects node A to broadcast next (this happens with probability of 50%). Node C receives m from A, but node D has to receive m from B. Thus again both A and B broadcast m.

To include an algorithm that constructs a *backbone* structure *prior* to the broadcast session, fig. 4.1 b) shows the output of the algorithm in [150] developed by Funke et al.. Funke's algorithm provably obtains one of the best approximation ratios to the MCDS. The algorithm is distributed and finds a CDS with size guaranteed to be within 6.94 from the optimum MCDS solution for UDGs. The algorithm has behavior similar to the Wan-Alzoubi-Frieder's algorithm from [149]. Funke's algorithm finds a solution CDS, which is a union of an independent set and a connected set in G. In the case of this example topology, the algorithm needs to select all the nodes in the network as its final solution.

Finally, fig. 4.1 c) show the optimal solution given a source node s_0. In this case the MCDS consists only of node B. The broadcasts of the source s_0 and node B would be sufficient to cover the entire network. All of the three algorithms considered above miss the optimum solution. Notice that by design all of the three algorithms approach the goal of reducing the number of broadcast nodes somewhat implicitly. RBS relies on expanding the area covered by the broadcast nodes quicker. CCS relies on *pruning redundant* transmissions, but the effectiveness of the selected broadcast nodes' transmissions is not considered as much. Funke's algorithm exploits interesting relationships between graph theoretic properties of the independent and connected dominating sets to provide a good theoretical guarantee for the final solution. However, it does not necessarily target the minimization of the number of broadcasting nodes locally at each step, or the maximization of the number of nodes that receive m for the first time.

Thus per all three algorithms the main utility of a broadcast node - the number of neighboring nodes that have not received message m yet - is not explicitly pursued or maximized, leaving room for inefficient broadcast decisions. In contrast, TSS's target is to *explicitly maximize the utility of each broadcast*: every broadcast node i should reach as many as possible neighboring nodes that have not received m yet. If there were a node j in the network whose transmission would potentially reach fewer such neighboring nodes, j should wait until i finishes its broadcast first. This principle and its careful distributed implementation allow TSS to find the optimum solution shown in fig. 4.1 c). Both nodes A and B receive the message m from s_0, however node B has two neighbors (nodes C and D) that have not received m at this time, and node A has only one neighbor (node C). TSS thus considers both A and B as broadcast candidates and assigns them respective tentative broadcast times. B has higher rank and receives higher priority than A. The broadcast time of B is earlier than the broadcast time of A. B broadcasts and at the time for broadcast assigned to A, A no longer has neighbors who have not received m. Following TSS policy A does not broadcast in this case. Thus TSS finds the optimal solution here: the broadcasts of the source node s_0 and B are sufficient.

Although the optimal performance of TSS in this extremely simple example is not an accident, the actual distributed design and implementation of TSS of course requires more care and justification. The next sections outline the algorithms behind TSS and some of the unexpected benefits of the scheme in highly dynamic mobile network scenarios.

4.1.2 Network Model

The network model consists of N equal-capability nodes with unique IDs, randomly distributed in a 2D plane. Though, the results below apply to 1D and to 3D networks, too. The transmission range of all nodes is r [meters]. Two nodes are referred to as 1-hop neighbors (or simply as neighbors) and can communicate directly if the Euclidean distance between them is less than R [meters]. Thus, the network is modeled as a Unit Disk Graph (UDG).

Two scenarios are considered regarding the MAC layer: a) a perfect MAC layer to isolate other effects (e.g., collisions, links asymmetry, etc.), so that the broadcast performance metrics reflect only the algorithmic efficiency; and b) packet loss (at the MAC and other network layers) to evaluate the performance of the algorithm in practical network settings, where collisions and links asymmetries, among other deleterious effects, are present.

The system operation is time-slotted, and the network nodes are assumed to be only coarse-grain synchronized. The latter is a standard assumption of many distributed algorithms and can be implemented in variety of ways. For instance, distributed, control-message-based coarse-grain synchronization in multi-hop wireless networks would suffice; such schemes have been studied extensively in the literature (e.g., [156]). Recent advances in radio technologies could also be utilized (e.g., [157]).

Definition 4.1. A *broadcast session* is the operation (including all related events) of delivering a message m, created at one node - the source - to all the other network nodes.

Definition 4.2. A *covered node* is a node that has already received the broadcast message in a prior transmission of the broadcast session.

To simplify notation, in what follows assume only a single message m needs to be propagated in the network, during the duration of each broadcast session. However, TSS can be trivially extended to handle the broadcast of multiple different messages simultaneously originating from different sources in the network.

The source node of a broadcast session is always covered. A node that has not received the broadcast message at a particular time, t, is referred to as an *uncovered node* at t.

Definition 4.3. The *residual coverage (RC)* of a covered node s ($s \in S$) at a particular time t, referred to as $RC(s)$, equals the number of its 1-hop uncovered neighbors at time t.

Define C as the set of all covered nodes at a particular time and Q as the set of nodes that have already transmitted the message at a particular time. Further, define $NE(s)$ as the set of all the neighbors of the node s, $s \in S$. Note that at any time, $Q \subseteq C \subseteq S$ and that $|S| = N$.

Finally, assume all nodes are cooperative.

4.1.3 Broadcast Solution Intuition

The problem of finding the most efficient broadcast scheme is equivalent to finding an approximation of the MCDS, and satisfying (1) - (5) above. Since finding the MCDS is an NP-hard problem [139], one needs to consider an appropriate heuristic, with the centralized greedy algorithm being one such an alternative (e.g., [158]). The basic idea of the greedy algorithm finding the MCDS is to repeatedly select nodes for transmission, such that in each round a node whose transmission covers the largest number of uncovered nodes is selected. Thus, each transmission “removes” the largest possible number of nodes from the set of uncovered nodes and, eventually, results in covering the whole network with a minimized number of transmissions. However, such a centralized greedy algorithm violates the requirement (4) of the previous section. Consequently, a particular distributed heuristic is discussed below, which approximates the operation of the centralized greedy algorithm. The operation of this distributed greedy heuristics relies on each node "scheduling" its transmission based on the value of its RC - the larger the value of RC, the sooner the node is scheduled to transmit.

To explain the operation of the proposed distributed heuristic, consider first the operation of the centralized greedy scheme. Start with the initial set of covered node $C = \{s_0\}$ and $Q = \emptyset$. The source node transmits first, covering its neighbors: $C = \{s_0 \cup NE(s_0)\}, Q = s_0$. An "oracle" greedily chooses a node, s_1, from the set $C \setminus Q$ with the largest RC value to broadcast next; i.e., $\forall s \in (C \setminus Q), RC(s) \leq RC(s_1)$. After s_1 transmits, $C \leftarrow C \cup NE(s_1)$ and $Q \leftarrow Q \cup s_1$. Then, repeatedly, the next node with the largest RC is selected to transmit from the set $C \setminus Q$, until all the network nodes are covered; i.e., until $C = S$, at which time the algorithm terminates. The total number of transmissions in a broadcast session equals $|Q|$ at the algorithm's termination time. Furthermore, the choice of node s_i to transmit in the i-th iteration allows maximizing the number of covered nodes during the i-th transmission. This intuitively only *tends* to minimize the total number of transmissions during the operation of the algorithm. The algorithm does not guarantee such a minimum, as in some cases choosing a node with smaller RC value first could, in fact, result in finding nodes with much larger RC values later, reducing the overall number of transmissions.

Although in [158], the authors discuss the inefficiency of a greedy scheme in finding MCDS in general graphs, the above centralized greedy heuristic finds on the average a rather close approximation of a MCDS in UDG as demonstrated in section 4.1.5. Of course, the challenge, similarly to other efficient MCDS approximation schemes, is to implement the “oracle” in a distributed manner; i.e., ordering nodes’ transmissions based on their RC values, while utilizing only local topological information. Surprisingly this path towards optimizing broadcast in wireless networks has not been investigated in works on the problem, prior to ([155]).

The following distributed *Time Sequence Scheme* (TSS) approximates the centralized greedy transmissions’ order in time, by allowing nodes with larger *RC* values to transmit before nodes with smaller *RC* values.

TSS’s blueprint is given as follows:

- Source s_0 transmits message m and covers its neighbors.
- Each node i receiving m for the first time marks itself as covered and computes $RC(i)$. The local *RC* computation by nodes is discussed below.
- Next, node i runs **Algorithm** (4.2) to schedule itself for later transmission time-slot T_b depending on $RC(i)$.
- If node i is scheduled to transmit in some time-slot T_b, i computes $RC(i)$ in the beginning of T_b. If the residual coverage of i has decreased (but is still positive) since the time-slot in which i has scheduled itself, i reruns **Algorithm** (4.2) and schedules itself for a new, later transmission time-slot. Else, still in T_b and prior to broadcast, i checks whether any of its 1-hop neighbors are scheduled to transmit within T_b as well. If more than one neighboring nodes are scheduled for T_b, the node with the largest *RC* transmits in T_b. The rest of the neighboring nodes schedule themselves to transmit in the next time-slot.

Next, we describe the details of the time sequence T'’s structure as generated by **Algorithm** (4.1). We also discuss the scheduling **Algorithm** (4.2) , and how it exploits the structure of T to rank, prioritize and order transmissions in time.

4.1.4 Time Sequence, Structure and Schedules

The timing of nodes' transmissions is enforced by the particular structure of the time sequence T of time-slots. Each time-slot $x, 0 < x \leq |T|$, is associated with a specific RC threshold τ_x. Only nodes with RC values greater or equal to τ_x are allowed to transmit in time-slot x.

4.1.4.1 Defining the Time Sequence

What is the rationale for determining the RC threshold τ_x at each time-slot x? Consider the following naïve scheme, which attempts to order the transmissions of the nodes, so that nodes with larger RC transmit first. Let T be a sequence of time-slots and assume $\tau_x < \tau_{x+k}, 0 < k \leq |T| - x$. Namely, each subsequent time-slot is associated with a strictly lower threshold value of RC than the previous time-slot's threshold. Upon receiving a broadcast message, node i marks itself as covered, determines $RC(i)$, and schedules itself to transmit in a future time-slot T_b. Since i can only schedule itself for a time-slot T_b such that $\tau_b \leq RC(i)$, the higher $RC(i)$ the earlier the scheduled T_b. That is, nodes with higher RC would tend to broadcast earlier than nodes with lower RC. This simple scheme does not take into account the fact that as scheduled nodes transmit, for instance in time-slot T_b, the set of newly covered nodes may contain nodes with RC values larger than τ_b. In other words, the time-slots following T_b cannot be used to time-order the transmissions of such newly covered nodes, since these nodes'

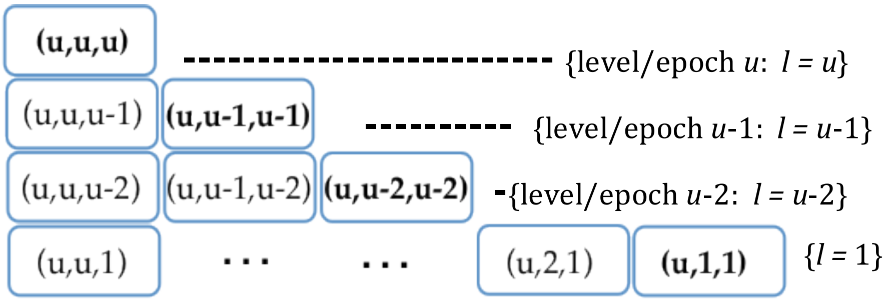


Figure 4.2: Time Sequence T: output of **Algorithm** (4.1) with input u, arranged in epochs to implement threshold resets required for node scheduling.

RC values are greater than the thresholds of all time-slots following T_b.

To address this problem, the time sequence T utilizes **Algorithm** (4.1) , so that T contains repeated reordering of time-slots within epochs (also referred to as levels interchangeably). Each epoch now contains a sequence of time-slots, and each subsequent time-slot within an epoch has *RC* threshold strictly lower than the previous time-slot's threshold. However, at the beginning of each epoch, the *RC* threshold is reset: the first time-slot in each epoch has *RC* threshold equal to the *RC* threshold τ_1 of the first time-slot in T. Suppose $\tau_1 = u$. Figure 4.2 shows the structure of the resulting time-sequence T, which is the output of **Algorithm** (4.1) with input u.

For example, consider the case of $u = 4$. At the top level of T (level 4) only nodes with $RC \geq 4$ are allowed to transmit. In the next level (level 3), first nodes with $RC \geq 4$ and then nodes with $RC \geq 3$ will be allowed to transmit. In level 2, first nodes with $RC \geq 4$, then nodes with $RC \geq 3$, and finally nodes with $RC \geq 2$ will transmit. In the last level, first nodes with $RC \geq 4$, then nodes with $RC \geq 3$, then nodes with $RC \geq 2$, and finally nodes with $RC \geq 1$ (all nodes with at least one uncovered neighbor) will be allowed to transmit.

To unambiguously label each time-slot, instead of using the threshold value only, we use a vector of three values: (upper, middle, lower). The upper is simply equal to u: the maximum value of the threshold, which is associated with the first time-slot in T. The lower is the number of the level/epoch and equals the *RC* threshold of this epoch. I.e. nodes with *RC* less than lower cannot transmit in this epoch. The middle is the threshold value of a time-slot. I.e. nodes with *RC* less than middle cannot transmit in this time-slot, but they may transmit later in this epoch, given their *RC* is greater than lower. Note that the values of lower and middle for all time-slots in T depend only on upper, which is equal to the parameter u. Hence, the number of timeslots $|T|$ in a broadcast session is a function of u. The parameter u is fixed and set up administratively and network-wide at the time of network deployment.

The value u should be judiciously chosen. A too small value of u does not allow separating in time the transmissions of nodes with different values of *RC*, thus losing the ability to assign larger priority to nodes with larger *RC* values,

Algorithm 4.1: ConstructTSSequence

```
input: upper value $u$
output: ordered collection of vectors $T = \{T_u, T_{u-1}, \cdots, T_1\}$
$T \leftarrow \emptyset$
$upper \leftarrow u$
$middle \leftarrow u$
$lower \leftarrow u$
$T_1 \leftarrow (upper, middle, lower)$
$T \leftarrow T_1$
while $middle > 1$ do
    if $middle == lower$ then
        $lower \leftarrow lower - 1$
        $middle \leftarrow upper$
    else if $middle > lower$ then
        $middle \leftarrow middle - 1$
    end if
    $T_{next} \leftarrow (upper, middle, lower)$
    $T \leftarrow T \cup T_{next}$
end while
```

while a too large value of u results in many empty time-slots, thus leading to an unnecessarily long broadcast session (i.e. larger $|T|$ and delay). Section 4.1.5.2 addresses the choice of appropriate parameter u in more detail, discussing its effect on the scheme's performance.

Figure 4.3 shows the ordering of time-slots in time during the duration of a broadcast session. Each "edge" slot (in bold) demarcates the end of an epoch. The time-slot from the uppermost epoch occurs first, followed by the time-slots in the next, lower level. Also, in each level the time ordering of the time-slots is from left to right. The uppermost epoch (which contains a single time-slot) is associated with the largest threshold of RC (which is set to u), allowing transmissions only of nodes with RC value of at least u. The second epoch is associated with the RC threshold of $u-1$, allowing only nodes with RC values of at least $u-1$ to transmit. However, notice that, as the transmission in the first epoch might have revealed newly covered nodes with RC value larger than u, the second epoch contains two time-slots: the first allowing transmission of nodes with RC value of at least u, followed by a time-slot allowing transmission of nodes with RC value of at least $u-1$. This process continues until the last epoch (associated with RC threshold of 1 and containing u time-slots) allows ordered transmission of nodes with RC values of at least u down to nodes with RC values of at least 1.

4.1.4.2 Scheduling Over the Time Sequence

Given the time-sequence (TS) structure described above, each node locally schedules its time of transmission, after receiving broadcast message m so that, overall, nodes with higher RC transmit earlier than nodes with lower RC. The TS serves as a common reference for all nodes.

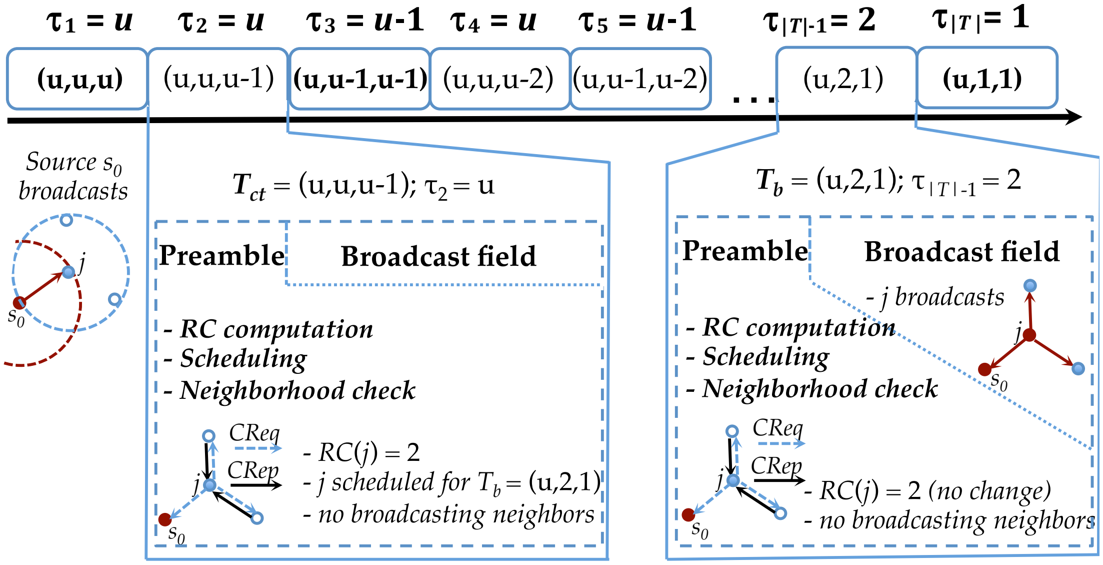


Figure 4.3: The output of **Algorithm** (4.1) with input u; temporal format of time-slots and the temporal blueprint of TSS. Only nodes with *RC* greater or equal than a time-slot's threshold τ_x can transmit in time-slot x. Time-slots in bold are called *edge slots*. Each time-slot comprises a *Preamble* followed by a *Broadcast Field*. Control messages are exchanged during the preamble of a time-slot. Nodes transmit a broadcast message during the broadcast field of a time-slot. T_{ct} is the current time-slot (j receives the broadcast message); T_b is a later time-slot for which j is scheduled using **Algorithm** (4.2) , based on $RC(j)$ computed during the Preamble of T_{ct}. In this case, $RC(j)$ has not changed during the time between T_{ct} and T_b; j broadcasts during the *Broadcast Field* of T_b.

The broadcast session begins when source node s_0 broadcasts message m. As m propagates throughout the network, any node j upon receiving m for the first time determines the current time-slot T_{ct} within the TS. This can be implemented as discussed in section 4.1.4.3 below.

The TS schemes' temporal flow proceeds as shown in fig. 4.3. Node j determines its residual coverage $RC(j)$, as described in section 4.1.4.3 below. After determining $RC(j)$, node j runs **Algorithm** (4.2) to schedule its transmission for a future time-slot. Given T_{ct} and $RC(j)$, **Algorithm** (4.2) schedules node j for a transmission timeslot, T_b, later in the broadcast session. Tb could be the next timeslot immediately after T_{ct} provided $RC(j)$ is large enough (i.e., $RC(j) > middle_{ct}$). Otherwise, **Algorithm** (4.2) attempts to schedule node j for a time-slot at the current level, if $RC(j) \geq lower_{ct}$. If the current time-slot is an edge slot (see fig. 4.2), **Algorithm** (4.2) attempts the next level of the time-sequence. Else, node j is scheduled to transmit at a later, lower level. In general, the larger $RC(j)$, the earlier is the level and the earlier is the scheduled transmission time-slot T_b within that level. If $RC(j) = 0$ the node is not scheduled for transmission at all.

It is important to note that the value of $RC(j)$ can change between the

Algorithm 4.2: SCHEDULEBROADCAST

```
input: $RC(j)$ of node $j$ to be scheduled, $(u_{ct}, m_{ct}, l_{ct})$ of the current time-slot $T_{ct}$
output: transmission time-slot $T_b \leftrightarrow t_b$
$rc \leftarrow RC(j)$
$upper \leftarrow u_{ct}$
$middle \leftarrow m_{ct}$
$lower \leftarrow l_{ct}$ // if $RC(j)$ is larger than the current value of $middle$, $j$ transmits
    in the next time-slot
if $rc > middle$ then
    $T_b \leftarrow T_{ct+1}$ // if $RC(j)$ is larger than $lower$, $T_b$ is in the current level de-
        pending on the value of $RC(j)$
else if $rc \leq middle$ and $rc \geq lower$ then
    if $(u_{ct}, m_{ct}, l_{ct})$ is $edge\_slot$ then
        if $lower > 1$ then
            $T_b \leftarrow (upper, rc, lower - 1)$
        else
            $T_b \leftarrow (upper, rc, 1)$
        end if
    else
        $T_b \leftarrow (upper, rc, 1)$
    end if
else if $rc < lower$ and $rc \geq 1$ then
    $T_b \leftarrow (upper, rc, rc)$ // if $RC(j)$ is even less than the of $lower$, $T_b$ is in a later
        level of the TS; the level is determined by $RC(j)$
end if
```

time at which j had scheduled itself for transmission and the beginning of j's scheduled-for-transmission time-slot T_b, due to transmissions of other nodes or due to mobility. This may render j inadmissible in T_b. To avoid transmission in an incorrect time-slot, j re-computes its RC value prior transmitting in T_b and checks if it still can transmit in T_b. If so, j transmits the message in T_b. Else, it reschedules itself by employing **Algorithm** (4.2) again with inputs $T_{ct} = T_b$ and the latest recomputed $RC(j)$.

4.1.4.3 Time Sequence Schemes Implementation Details

This section discusses possible implementations of the Time Sequence Schemes based on the algorithms described above. It also provide details regarding implementations of procedures such as residual coverage computation, determining the current time-slot, and checking if a node has the largest RC in its neighborhood.

As shown in fig. 4.3, each time-slot consists of a *Preamble* part immediately followed by a *Broadcast Field* part. The *Broadcast Field* is fixed to the maximum duration needed to transmit the broadcast message depending on the maximum message size. The *Preamble* is used to transmit short control messages between adjacent nodes, and its duration is small compared to the *Broadcast Field* length of the time-slot. The time-slot duration t_D equals the duration of the *Preamble* added to the duration of the *Broadcast Field* and is known to all nodes in the network at deployment time.

Determining the Current Timeslot:

The source node transmits the message m at the beginning of the *Broadcast*

Field of the first time-slot in the TS. The initial transmission's timestamp is piggybacked by the broadcast message. Upon receiving m, node j computes the current time-slot T_{ct} by subtracting the initial transmission timestamp from j's current local time and dividing the difference by t_D to obtain the number of elapsed time-slots since the beginning of the broadcast session. Knowing the generic TS structure as per fig. 4.2, a node is able to determine the vector of the current time-slot: $T_{ct} = (upper_{ct}, middle_{ct}, lower_{ct})$.

Residual Coverage Computation:

The *RC* value of a node is needed prior to the node's scheduling (during the *Preamble* of timeslot T_{ct}) or rescheduling (during the *Preamble* of timeslot T_b), as noted in section 4.1.4.2 above and shown in fig. 4.3. This is done by a locally executed protocol - a version of a "Neighbor Discovery" protocol - where *Coverage Request* (*CReq*) and a *Coverage Reply* (*CRep*) messages are exchanged between neighboring nodes.

This simple protocol can be further improved in a variety of ways. However, in section 4.1.5.2, TSS is simulated over a full network stack, demonstrating that a standard 802.11b MAC layer readily handles all the control messages generated; as the network density increases, the performance of the TS schemes remain robust (figures 4.8, 4.7, and 4.6).

The Naïve Time-Sequence Scheme

For clarity, first a basic TS-based broadcasting scheme is outlined next. Upon network deployment, all nodes run **Algorithm** (4.1) to construct the TS. After a node transmits the broadcast message, all of its previously uncovered neighbors that receive the message mark themselves as covered, compute their *RC*, and run **Algorithm** (4.2) to schedule their transmission time-slots. Just before its scheduled time to transmit (during the *Preamble* of T_b), node j re-computes and updates its $RC(j)$, as per fig. 4.3. After the update, if the $RC(j)$ has not decreased, j broadcasts during the *Broadcast Field* of T_b; if $RC(j)$ has decreased (but $RC > 0$), the node determines its new time-slot assignment by re-running **Algorithm** (4.2). If, at any time, the computed RC value of a node equals 0, the node will never be scheduled for transmission in this broadcast session. This basic scheme is referred to as the Na¨ve Time Sequence Scheme (NTSS). As the name indicates, the NTSS possess some important deficiencies, which will be cured by the other variant, TSS, of the scheme presented next.

The Time-Sequence Scheme (TSS): Neighborhood Check

TSS operates as NTSS, but incorporates a 1-Hop neighborhood check within the scheduled-for-transmission time-slot T_b. To accommodate that, each timeslot's *Preamble* is split in two parts: *Preamble*1 and *Preamble*2. Suppose node j is scheduled to transmit in T_b. During *Preamble*1, j computes its RC, by sending *CReq* and receiving *CRep* packets. Next, j checks during *Preamble*2 of T_b whether any of its 1-hop neighbors are scheduled to transmit within T_b as well. This check does not necessitate any additional transmissions. Node j can determine whether a particular neighbor i is scheduled to transmit in T_b, if j has received the *CReq* message from i during *Preamble*1. If more than one neighboring nodes are scheduled for T_b, the node with the largest RC is selected to transmit in T_b. In our implementation of TSS this is done as follows. Immediately after *Preamble*1, node j picks a random time within *Preamble*2 to send a $RCPacket_j$ containing $RC(j)$ (as found by j in *Preamble*1) and j's ID. If j receives $RCPacket_i$ from a neighboring node i and if $RC(i) > RC(j)$, j does not send its

$RCPacket_j$ and reschedules itself for broadcast to the next time-slot. Otherwise, j sends its $RCPacket_j$. Since the set B_N of all neighboring nodes broadcasting in the same time-slot T_b follows this protocol, it is easy to check that at the end of *Preamble*2 all but the node j^* with highest RC in B_N broadcasts in T_b. The nodes in $BN \setminus j^*$ are rescheduled for the next time-slot T_{b+1}.

This 1-Hop check avoids redundant transmissions whereby neighboring nodes i and j broadcast m to the uncovered nodes in their respective neighborhoods and the shared neighbors of i and j receive m redundantly both from i and j. This severely degrades the performance of NTSS (see section 4.1.5).

4.1.4.4 Sample Execution of TSS

Consider the network of nodes shown on fig. 4.4. Suppose $u = 4$ and **Algorithm** (4.1) constructs the TS as shown in fig. 4.2. The sample run of TSS is shown in Table (4.1), with the resulting network coverage depicted in fig. 4.4. In the first time-slot, the source node s_0 transmits the message m. Nodes A, B, C, and D receive m, mark themselves as covered, compute their RC, and schedule themselves to broadcast. At **Step 1**, according to Algorithm (4.2), since node B has 3 uncovered neighbors it is scheduled for time-slot (4,3,3). Node A is scheduled similarly for (4,2,2); C and D for (4,1,1). At **Step 2**, during the third time-slot (4,3,3) node B transmits the message. Nodes G, H, and F become newly covered and are added to the scheduled nodes list: F is scheduled for (4,1,1); G and H have $RC = 0$ and are not scheduled. Node B is removed from the list. At **Step 3**, in the sixth timeslot (4,2,2), node A checks its RC. Since its RC remains the same, node A transmits the message. Nodes E and K become newly covered. Node K has two uncovered neighbors: I and J. K is scheduled for time-slot

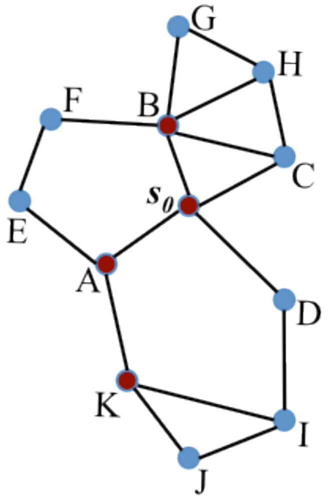


Figure 4.4: A network topology example, where the TS-based scheme picks $\{s_0, B, A, K\}$ for broadcast and forms a MCDS of the particular network graph. All nodes are covered after the broadcast session completes. The TSS steps are given in Table (4.1).

(4,2,1). Finally, at **Step 4**, during the preamble of the ninth time-slot node K has not changed its RC. K transmits in this time-slot. All nodes are covered at this point. Hence, C, D, and F do not transmit.

Note that the TS structure allows **Algorithm** (4.2) to give priority to nodes with higher RC, which could be covered and scheduled later during the broadcast session. For instance, because of its larger RC value, node K transmits before either of nodes C, D, or F, in spite of the fact that C, D, and F received the broadcast message and were scheduled earlier than K.

Ultimately, this eliminates the transmissions of nodes C, D, and F. Fig. 4.4 shows the network state after the broadcast session is completed. In this example, the MCDS (in red) equals the nodes selected by the TS scheme.

Table 4.1: TS Schemes Example Run

Scheduled Nodes	RC/Scheduled Timeslot
A	2 / (4,2,2)
B	3 / (4,3,3)
C	1 / (4,1,1)
D	1 / (4,1,1)

Step 1

Scheduled Nodes	RC/Scheduled Timeslot
A	2 / (4,2,2)
C	0 / (4,1,1)
D	1 / (4,1,1)
F	1 / (4,1,1)
G	0 / -
H	0 / -

Step 2

Scheduled Nodes	RC/Scheduled Timeslot
C	0 / (4,1,1)
D	1 / (4,1,1))
F	0 / (4,1,1)
G	0 / -
H	0 / -
E	0 / -
K	2 / (4,2,1)

Step 3

All nodes are covered after timeslot (4,2,1). The time-sequence is exhausted at (4,1,1), where as well no nodes have $RC > 0$, and the algorithm terminates.

Step 4

4.1.4.5 Properties of the TS-based Schemes

Correctness of the TS-based Schemes

In the following proof of correctness, the network topology is assumed to be static during the broadcast session and the MAC layer to be perfect. The network graph is assumed to be connected. For an arbitrary mobility model even flooding cannot ensure full network coverage; similarly, this is true if package loss probability is positive. Also, if the underlying graph of a static network is disconnected no broadcast algorithm can ensure full network coverage.

The performance evaluation of the TSS in the next section, demonstrates that even in the presence of high mobility and potential packet loss, the scheme achieves full or almost full network coverage.

Theorem 4.1. *The TS-based schemes terminate in finite amount of time and guarantee full coverage of the network.*

Proof. Per **Algorithm** (4.2), for all TS schemes, a node is not scheduled to transmit unless its RC is strictly greater than zero. Whenever a node n transmits, all of it neighbors receive the broadcast message and are marked as covered. Hence, the RC value of node n decreases to zero, and node n is not admissible in any future time-slot. Therefore, a node does not transmit more than once during the execution of the algorithm. Since the number of nodes in the network is finite, the algorithm terminates in a finite number of steps.

Now, suppose that after the termination of a TS-based algorithm there is at least one node, D, that is not covered. Since the network graph is connected, there exist at least one path from the source node, S, to the destination node D. Because D has not received the message, there are at least two neighboring nodes X and Y along this path, such that X has received the message and Y has not received the message (note: X might be S and Y might be D). Therefore, since Y has not been covered, $RC(X) \geq 1$ at algorithm's termination. This is a contradiction. Per both TS schemes, node X computes $RC(X)$, after receiving the broadcast message. Per **Algorithm** (4.2) (lines 5-17), as long as $RC(X) > 0$, X is always scheduled to transmit in a later timeslot T_b.

Thus, the TS schemes cover all the network nodes. □

Algorithm (4.1) *Complexity*

Let T be an ordered collection of vectors $\{T_x, T_{x-1}, \cdots, T_1\}$, where $T_k = (u_k, m_k, l_k)$, u_k, m_k, and $l_k \in \mathbb{N}^+$, and where the parameters u, m, and l are equal to the *upper*, *middle*, and *lower* values. Let T be the output of **Algorithm** (4.1) with input u.

Theorem 4.2. *The time complexity of* ***Algorithm*** (4.1) *is* $O(u^2)$*; and the length of the generated time sequence is* $x = |T| = u(u+1)/2$.

Proof. The output, T, of **Algorithm** (4.1) is arranged in an isosceles triangle with sides u as shown in fig. 4.2. The triangle consists of u levels, where the last level (level 1) comprises u vectors. Invariantly, the number of vectors at the i-th level equals $1 + u - i$. The total number of vectors is then:

$$\sum_{i=1}^{u} = \frac{u(u+1)}{2} \tag{4.1}$$

At each iteration of **Algorithm** (4.1) there is exactly one vector generated. Hence there are $O(u^2)$ iterations and $x = |T| = u(u+1)/2$. □

As discussed in section 4.1.5, the value of u is rather low ($u \approx 7$) in practice and does not depend on the network density. Hence, **Algorithm** (4.1) is not computationally expensive and could be run on resource-constrained nodes.

Greedy Transmission priority

Let $A = (S, \mathcal{I})$ be a set system with ground set $S = \{j : 1 \leq j \leq N\}$, the set of all network nodes. Each node j has weight equal to $RC(j)$ at a given time t. As time advances from time-slot to the next, the values $RC(j)$ may change. Let $\mathcal{I} = \{S_1, S_2, \cdots, S_{|T|}\}$ be the collection of subsets of S, where $S_k = \{j : RC(j) \geq m_k \geq l_k \geq 1\}$, for given vector $T_k = (u_k, m_k, l_k) \in T$. Note that $|\mathcal{I}| = |T|$.

Definition 4.4. A network node j is called admissible in vector T_k, iff $j \in S_k$.

Let $x = |T| = \frac{u(u+1)}{2}$. Every x consecutive time-slots (ordered in time as in fig. 4.3) are mapped one-to-one to the vectors in the ordered collection T above. That is, each time-slot, t_k is uniquely associated with a vector in T: $t_1 \leftrightarrow T_x = (u, u, u), t_2 \leftrightarrow T_{x-1} = (u, u, u-1), \cdots, t_x \leftrightarrow T_1 = (u, 1, 1)$.

Formally, a time-sequence TS is the ordered collection of the time-slots together with their corresponding vectors in T. A network node is admissible in time-slot t_k if it is admissible in vector T_{x+1-k}. This association "wraps around"; i.e., in general for $k \geq 1, t_k \leftrightarrow T_{d+1-k}$, where $d = \lceil \frac{k}{x} \rceil$.

Revisited Definition 4.1. A broadcast session consists of all the events, starting from the transmission of the message m by the source node and ending when the broadcast algorithm terminates after $|T|$ time-slots.

Broadcast Rule: At every time-slot t_k, a node considers transmitting only if it has not transmitted earlier in this broadcast session and if during the *Preamble* of t_k, it is admissible in T_{x+1-k}.

It is easy to verify that **Algorithm** (4.2) complies with the *Broadcast Rule*. For instance, if $middle_{ct}$ of the current TS time-slot is lower than a node's RC value, **Algorithm** (4.2) schedules the node's transmission for the next immediate time-slot. Otherwise, a further future time-slot is assigned to the node. A node is never scheduled to transmit if its $RC = 0$ (i.e., it is not admissible in any time-slot).

Definition 4.5. *Node Ordering*: $j \leq i, \forall i, j \in S$, iff $RC(j) \leq RC(i)$.

Definition 4.6. A *minimal admissible element* $\min(T_k)$ in vector T_k is an element

$j \in S$ such that $j \leq i$ for all i admissible in vector T_k.

Consider the ordered collection T^q of vectors at level $q, q \in \{1, \cdots, u\}$, of T.

Definition 4.7. $inf(T^q)$ is the smallest element among all minimal admissible elements in the vectors at level q.

Let L be the sequence $(inf(T^u), inf(T^{u-1}), \cdots, inf(T^1))$. Note that the sequence L is decreasing, from Definitions (4.5), (4.6), and (4.7) applied to $\mathcal{I}$.

Since L is decreasing, given the TS structure and *Broadcast Rule*, covered nodes with low *RC* values would not be able to transmit (i.e., be admissible) in earlier higher levels of the time sequence TS, but will potentially be admissible in later, lower levels. And reversely, only nodes with larger *RC* values would be admissible and be able to transmit in earlier higher levels of the time sequence. Next, note that a similar observation holds for the time-slots within each level of the TS.

Let M be the sequence $(min(T_i), min(T_{i-1}) \cdots, min(T_1))$, where vectors $T_i, T_{i-1}, \cdots, T_1$ are in collection T_q for a given level q. The sequence M is decreasing again from applying Definitions (4.5), (4.6), and (4.7) to $\mathcal{I}$.

Since M is decreasing, within level q of the TS, nodes with smaller *RC* values would be admissible only in later timeslots at level q, and nodes with larger *RC* values would be admissible earlier at level q.

In summary, the structure of the TS (implemented via **Algorithm** (4.1)) in conjunction with the *Broadcast Rule* (implemented via **Algorithm** (4.2)) has admissibility property allowing for repetitive assignment of larger transmission priority to nodes with larger *RC* values compared to nodes with smaller *RC* values. By virtue of the admissibility property of the *Broadcast Rule*, the greedy ″oracle″ scheme is emulated approximately.

4.1.5 Performance Evaluation and Comparison

The performance, defined by a number of metrics, of various broadcast algorithms in four distinct network topology models is investigated next. For a static network topology, consider the case of a perfect MAC-layer (no packet loss due to collisions) is considered first. Next, NTSS and TSS are implemented in a full network stack, discrete event simulator (JiST/SWANS [119]), where packets may be lost at different network layers (e.g. due to collisions, noise etc.). Finally, the the algorithms' performance is compared under two types of realistic mobile models: one generating independent mobility patterns of the nodes; and another generating correlated (group) mobility patterns.

The performance of the TS-based schemes is compared against the most efficient schemes found in the technical literature to date. In particular, the simulated algorithms are the *RBS* ([154]). In [154] the authors show that RBS outperforms a few well-known broadcast algorithms such as the *Edge Forwarding* [159] algorithm, for example. The more recent *CCS* ([136]) is also simulated. Another broadcast protocol implemented for comparison here is the *Bordercast*: the route discovery mechanism in the *Zone Routing Protocol* (ZRP) ([119]). *Bordercast* relies only on local topological information to select the nodes, which forward the broadcast message. Per *ZRP* a zone of node *A* in the network includes all nodes that are within *k* hops from *A*. Border nodes are those nodes in the zone whose minimum hop distance from A is exactly k. According to the Bordercast algorithm, the goal is to cover most efficiently the set of border nodes in its zone.

To include an algorithm that constructs a backbone structure prior to the broadcast session, Funke's algorithm from [150] was selected, as it arguably provides one of the theoretically closest constant approximation ratio to MCDS: 6.94. For comparison, Liu's influential algorithm from [160] is also simulated - this is a *node forwarding* algorithm that relies on 1-Hop positional information.

In all the experiments, unless otherwise indicated, the simulation area is a $200[m] \times 200[m]$ square; the inner square area is of dimensions $(200 - R)[m] \times (200 - R)[m]$ to avoid edge effects. $R[m]$ is the transmission radius of all nodes and is set to $R = 25[m]$. The number of nodes in the network varies from 200 up to 3000 nodes to investigate the schemes' performance at different node densities.

4.1.5.1 Static Network Topology with Lossless MAC

In this case the broadcast schemes performance reflect only the algorithmic efficiency, without the effect of collisions or noise in the environment.

The number of transmission (i.e., "transmission complexity"), equivalent to the number of broadcasting nodes during a broadcast session is a crucial metric for an efficient broadcast algorithm, and is investigated in figures 4.5 **top** and **bottom**. The performance of the centralized Greedy algorithm is plotted in 4.5 **bottom** for comparison with the TSS. As expected TSS emulates well the centralized Greedy algorithm and both generate similar number of transmissions. Interestingly, without utilizing positional information TSS achieves about 15% lower number of transmissions compared to RBS and completes the broadcast session with approximately 67% fewer transmissions than CCS, while utilizing only 1-Hop topology knowledge. As an additional benchmark for transmission complexity, the figure also shows the number of transmissions using the Linear Hexagon Coverage technique, which provides a very close approximation to the minimum number of transmissions needed to cover the entire network area, assuming sufficient node density (fig. 4.5 **bottom**). The upper bound of transmission complexity in fig. 4.5 **top** should be interpreted as the maximal number of transmissions that would be required by any broadcast algorithm that avoids duplicate coverage of nodes and, hence, is density-independent. More detailed discussion of these bounds is found in Appendix D.1 and D.2.

4.1.5.2 Full Network Stack Simulation

TSS and NTSS are implemented in the JiST/SWANS network simulator ([119]) to evaluate the effect of collisions, noise, fading, link asymmetries, etc. inherent in practical network scenarios.

The simulation parameters are given in Table (4.2). Figures 4.6, 4.7, and 4.8 show the performance of the TS schemes in terms of transmission complexity, delay, and fraction of network covered. Since some packets may be lost, full network coverage cannot be 100% guaranteed. However, the performance of the TSS scheme is rather robust as node density increases, despite more load at the MAC layer is present in the form of control messages. For comparison, the state-of-the-art RBS ([154]) and CCS ([136]) algorithms, discussed above, are also implemented in JiST/SWANS.

Transmission Complexity:

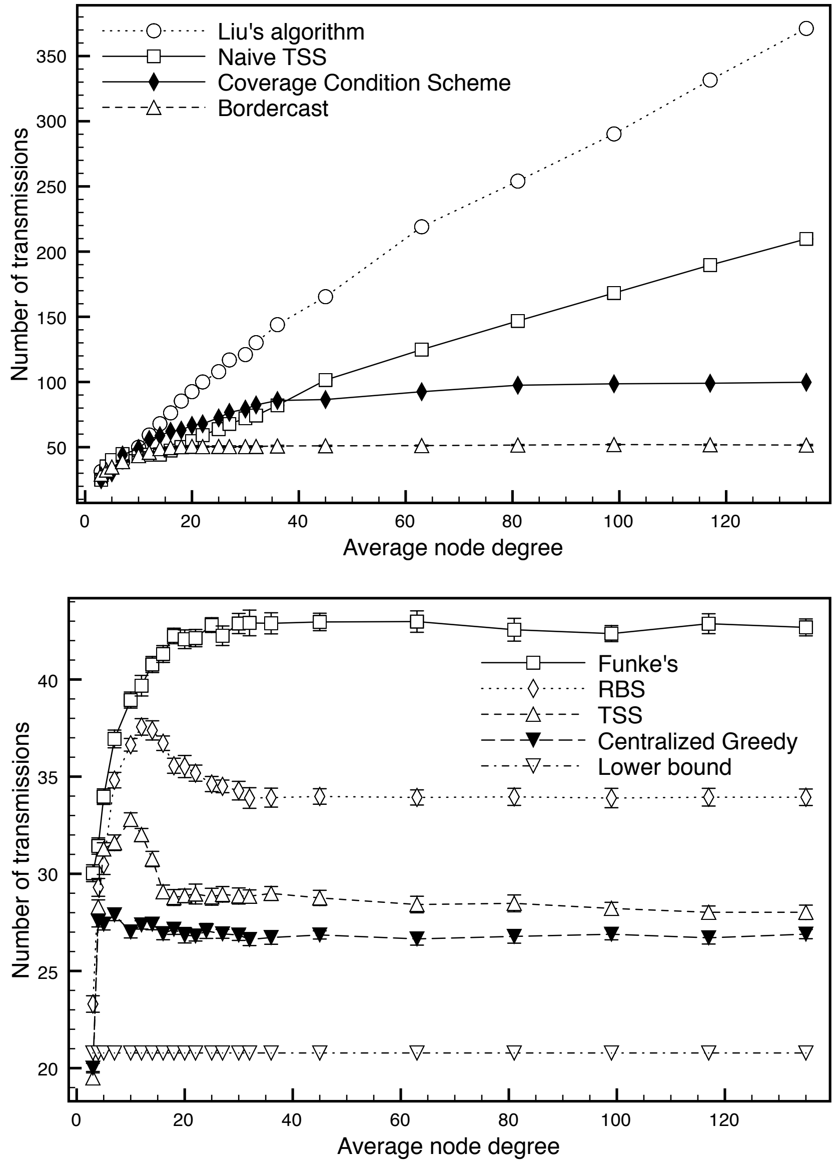


Figure 4.5: The number of transmissions in a static collision free network (full coverage) of eight different broadcast algorithms.

Table 4.2: Parameters of the JiST/SWANS Simulation

Simulator	JiST/SWANS v1.0.6
MAC layer	IEEE 802.11b
Radio frequency	2.4GHz
Propagation model	Free space
Packet size	64 - 7081 [bytes]
R	≈ 22[m]
Area	Square: 200[m] × 200[m]
Number of nodes	[150-1000]
Time-slot duration	TSS/NTSS: 150[ms]

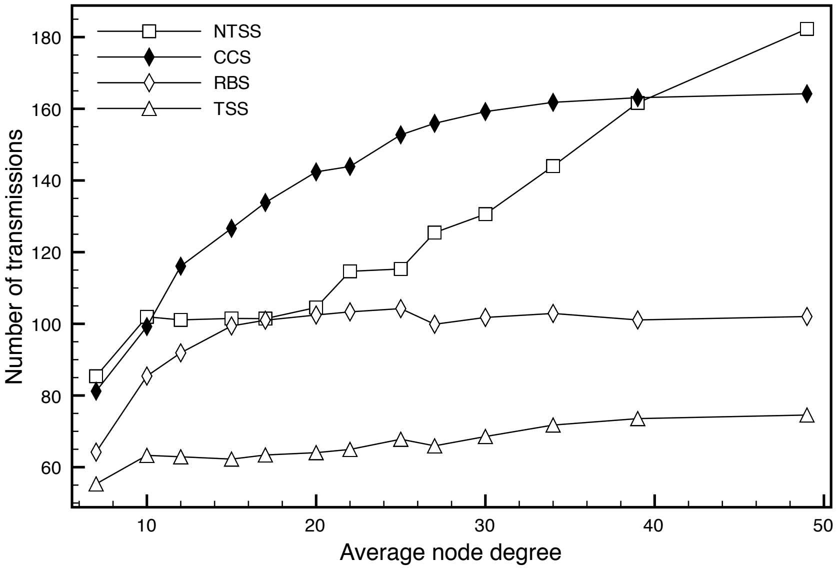


Figure 4.6: Number of retransmissions (full network stack simulator).

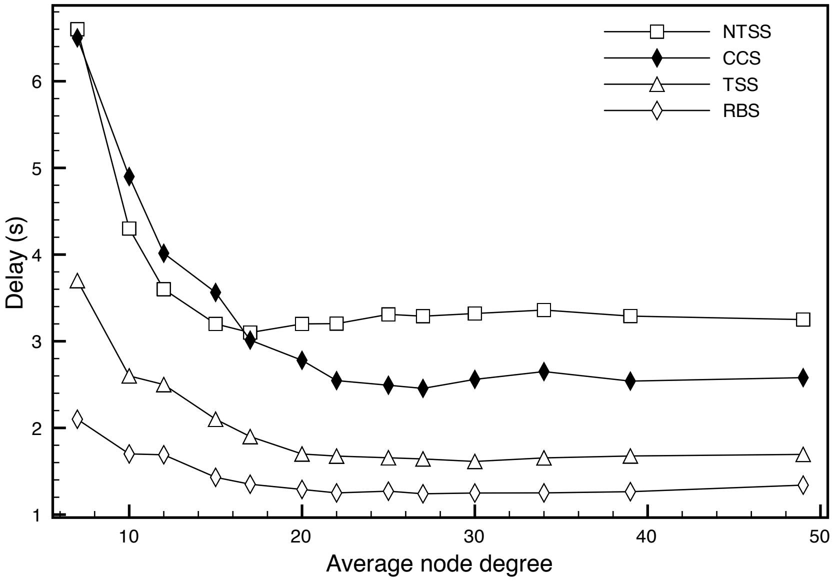


Figure 4.7: Delay (full network stack simulator).

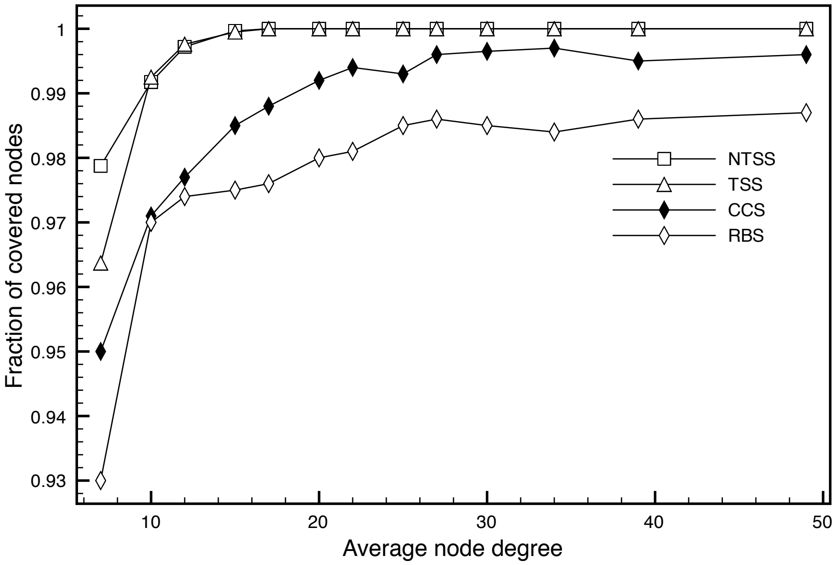


Figure 4.8: Network coverage (full network stack simulator).

The simulation results in terms of number of transmissions (fig. 4.6), both for the TSS and NTSS algorithms, resemble their respective performance trends in the case of ideal MAC layer above. The number of transmissions is higher in fig. 4.6 since the average transmission radius is about 22[*m*] here vs. 25[*m*] in the ideal MAC layer case. Also, some broadcast messages may be lost, requiring further retransmissions. TSS generates 25-30% and 53% fewer transmisions than RBS and CCS respectively. Notice however that, simultaneously, TSS maintains higher network coverage (fig. 4.8): for instance, RBS leaves 25-30 uncovered nodes for networks of average node degree about 40.

Delay:

The delay - the time needed to complete a broadcast session - is presented in fig. 4.7. The delay performance of the TS-based schemes is obtained with parameter u set to the smallest value possible, so that the number of broadcast transmissions is still minimized. Decreasing u further would decrease the delay but would lead to more transmissions.

Setting the parameter u: On one hand, the number of time-slots would be lower if u is lower. However, smaller number of time-slots allows for more nodes with different RC's being admissible and able to broadcast in the same time-slot. Hence, nodes' transmissions greedy prioritization could be coarser and less efficient. Larger values of u lead to potentially better prioritization of nodes' transmissions. However, the probability that a time-slot does not contain any transmissions is increased, leading to unutilized time and higher delay (quadratic in u as shown in section 4.1.4.5). For instance, initial time-slots go by empty since their corresponding middle and lower values are too high compared to the nodes with highest RC. Thus, very few broadcasting nodes are admissible then.

Optimal value of parameter u is robust to network density variation: The optimal values of u for different network densities were determined via simulations (fig. 4.9). Observe that beyond average node degree of 15, the optimal u values for TSS remain essentially constant with respect to the network density. Increasing u further, does not lead to notable decrease of transmission complexity. Thus the upper value could be fixed prior to network deployment, resulting in transmission complexity and delay that are close to the TSS optimal performance. The results for TSS hint that a good practice would be to set u to the average node degree for networks of node degrees up to 7. For networks of larger node degrees, u is largely density-independent and may remain fixed to 7. In the implementation presented here, each node constructs a local copy of the TS that can be extended if at the last time-slot of the TS, the node schedules itself for transmission and its residual coverage is greater than 0. The delay of the broadcast (fig. 4.7) accounts for the number of time-slots in the time sequence at the node with the longest local time sequence (i.e. the time elapsed between the first and last transmission).

The delay of TSS is comparable but slightly larger than that of the RBS algorithm, which however relies on GPS information and leaves larger number of nodes in the network uncovered (fig. 4.8). The delay of the TSS scheme is about 50% lower compared to the CCS scheme, which utilizes only topology information.

Network Coverage:

The network coverage (fig. 4.8) of the TS-based schemes for lower node

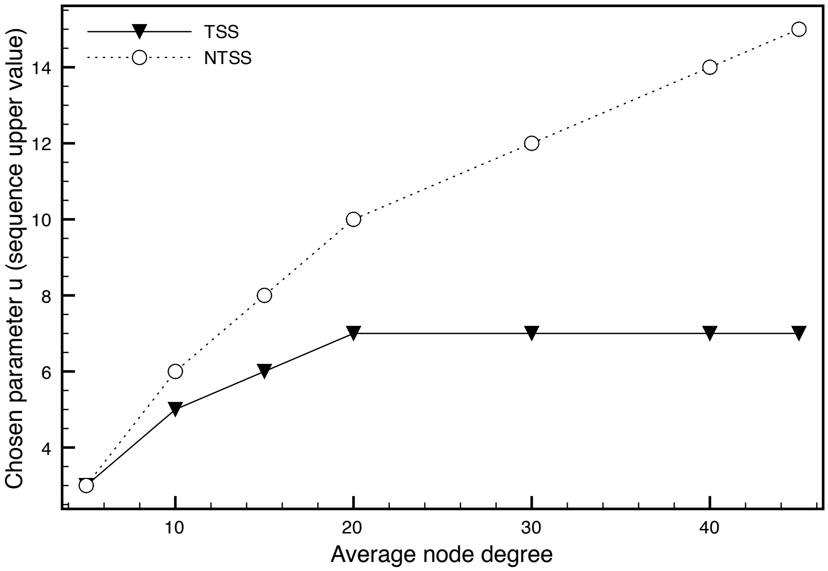


Figure 4.9: The values of the parameter *u* determined via simulations for NTSS and TSS, so that the trade-off between number of rebroadcasts and delay is optimized.

densities is ≈ 96% since disconnections in the network are possible. However, at higher network densities the TS-schemes performance remains robust and they achieve virtually *full* coverage. The results indicate that the MAC layer readily copes with the number of control messages, as density increases. For all network densities TSS maintains higher coverage than both RBS and CCS.

4.1.5.3 Dynamic Network Topology

In addition to TSS, two state-of-the-art online, dynamic broadcast schemes (i.e. RBS, and CCS) are simulated under two mobility models within JiST/SWANS. Under the *Gaussian-Markov Mobility Model* (*GMMM*) ([161], [162]) each node follows independent realistic trajectory of movement. Under the *Self-Similar Least Action Model* (*SLAW*) model from [163], subsets of the network nodes follow correlated paths.

Table 4.3: Parameters of the GMMM

Velocity and position update interval	0.2 [s]
Velocity standard deviation	0.75 [m/s]
Velocity mean	1-20 [m/s]
Alpha	0.75

Individual Movement: Gauss-Markov Mobility Model:

Per GMMM, time is split into time intervals (independent of the TS-based schemes time-slots). At the beginning of the k-th time interval, nodes' velocity is updated according to the following rule:

$$v[k] = \alpha v[k-1] + (1-\alpha)\bar{v} + (1-\alpha^2)^{\frac{1}{2}} z[k-1] \tag{4.2}$$

Here, $v[k-1]$ is the velocity (speed and direction) of a node in the $[k-1]$th, time interval; $z[k-1]$ is the observation of a Gaussian random variable at time interval $[k-1]$; $\bar{v}$ is the mean value of the velocity; and α is a parameter that determines the degree to which the current velocity at step k depends on the velocity at time interval $[k-1]$. As α approaches 1, nodes' motion becomes more constant; as α approaches 0 nodes' motion becomes more random. Table (4.3) summarizes the values of the parameters used in the simulation. The number of transmissions and the corresponding achieved network coverage by the four algorithms at different average speeds is shown in fig. 4.10 **top**. The TSS performance is robust since each node checks its residual coverage at least twice (at the time a broadcast message is received and prior to transmission in the scheduled-for-transmission time-slot). Notice also that TSS is partially resilient against temporal disconnections of nodes from the network due to mobility.

More specifically, suppose in the *Preamble* of time-slot T_{ct}, j has a relatively smaller number of neighbors compared to the average node degree in the net-

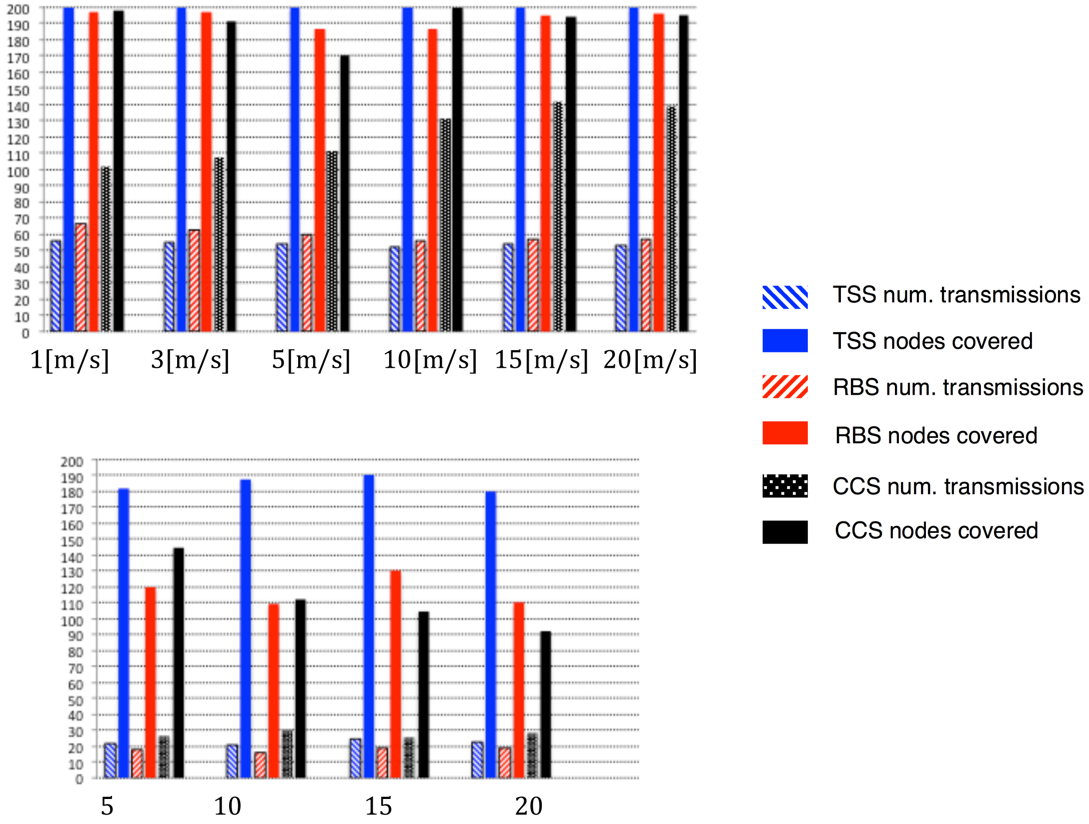


Figure 4.10: Mobile network: top) Gaussian-Markov Mobility Model: N = 200 nodes. Num. transmitting and covered nodes (non-solid and solid color respectively); bottom) Self-similar Least Action Model: N = 200 nodes; x-axis: varying number of fractal waypoints 5,10,15, and 20.

work. Respectively, j has higher chance of being temporarily disconnected from the network. In this case, it is likely that $RC(j)$ would also be lower in comparison to the RC of other nodes in the network. Then, node j is scheduled for a broadcast time-slot T_b that is late in the TS. This gives j more time to potentially move in areas with higher number of neighbors and increase $RC(j)$. Furthermore, even if meanwhile j becomes disconnected from the network at some time-slot between T_{ct} and T_b, that would not affect j's decision to transmit or not during T_b. Only $RC(j)$ in the *Preamble* of T_b determines whether j would transmit or not in T_b.

In contrast, suppose node j has larger than average number of neighbors during the *Preamble* of time-slot T_{ct}. In this case it is likely that $RC(j)$ would also be larger in comparison to the RC of other nodes in the network. Hence, j would be scheduled to transmit sooner in the TS, thus reducing the chance that the number of j's neighbors would decrease due to mobility.

As a result, TSS generates $\approx$ 15% fewer transmissions than the position-aware RBS while achieving full network coverage. RBS leaves $\approx$ 5% (8-10 nodes) uncovered. Furthermore, TSS generates 50-60% fewer transmissions than CCS, while the latter leaves up to 15% of the network nodes uncovered

Group Movement: Self-Similar Least Action Model

Since independent individual node movement may be unrealistic for certain scenarios (e.g. hikers, tour groups, military platoon, etc.), a group mobility model is also simulated here. The model is based on the *Self-Similar Least Action Model* (*SLAW*) model described in [163]. SLAW aims specifically at modeling realistically human mobility statistical features and has been verified to match a number of traces generated by real people's mobility. The simulation parameters (such as β, Levy Exponent, Waypoint Ratio, Hurst Value, and node velocities) used here were borrowed from [163]. Table (4.4) lists the parameters' values. Figure 4.10 **bottom** depict the number of transmissions and the corresponding network coverage for different number of fractal waypoints governing the clustering in the network. TSS achieves at least 90% network coverage. The performance of the remaining broadcast algorithms degrades substantially since the network tends to be temporally partitioned. Here, each transmission

Table 4.4: Parameters of the SLAW model

Pause time	10-120[s]
β	1
Cluster range	27.5[m]
Distance weight	3
Nodes velocities	0.5 - 2 [m/s]
Waypoint ratio	5
Hurst value	0.75
Number of fractal waypoints varied	[5-20]

covers larger number of nodes (due to clustering) and fewer transmissions are required to cover the network. TSS generates more transmissions than RBS, however it covers substantially larger fraction of the network nodes.

4.2 Other Broadcast Schemes

The problem of efficient broadcasting has been extensively studied in the technical literature. The initial simple concept of flooding (e.g. [11]) evolved into more sophisticated schemes through building optimal network subgraphs. Among the major shortcomings of pure flooding are the large transmission complexity and the resulting notorious *broadcast storm* [11], where the network suffers from severe contention due to the number of transmissions. The *Scalable Broadcast Algorithm* [164] alleviates somewhat this problem utilizing 1-Hop neighbor information. All through, the main algorithmic challenge has been to reduce the number of transmissions needed to reach all network nodes.

More recent work on flooding offers a different and ingenious approach to circumvent the broadcast storm problem. Ferrari et al. ([138], [165]) design a system that harnesses constructive interference of concurrent neighboring transmissions and achieves remarkably low broadcast latency. Finding an approximation CDS to the minimum connected dominating (MCDS) set first, as done by the proposed TSS scheme, can significantly reduce the number of transmissions generated by such novel flooding-based algorithms. Constraining the flood to the nodes in the CDS could lead to significantly reduced energy expenditure, for instance, as observed in [137]. Consequently, these flooding-based approaches are orthogonal to the application of schemes such as TSS in finding approximations to the MCDS.

A different approach is taken by probabilistic broadcast protocols that associate some (re)transmission probability to each node receiving the broadcast message. Schemes exploring such mechanisms were suggested in ([140], [141], [142], [143], [144], [145], [146], [147], and most recently [166]). The interest in probabilistic broadcasting schemes is due to their inherent low transmission overhead, low processing complexity, and high tolerance to frequent and rapid topological changes. Balancing these benefits, though, is the disadvantage of inability to guarantee full network coverage and, still, the presence of many redundant transmissions.

In contrast, deterministic broadcast algorithms innately guarantee full network coverage (assuming ideal MAC layer). In the deterministic scheme of *Multipoint Relaying* proposed in [167], the set of retransmitting neighbor nodes is reduced from the set of all neighbors to the minimum subset of neighbors that cover the same area as that covered by the original set. This approach is an example of the *minimum forward-node set* strategy, and works such as [160], [168],

[159], and [169] provide approximate solutions to this NP-hard problem. To avoid the transmission of the list of forwarding nodes along with the broadcast message, the technique of *self-pruning* (e.g. [170], [171]) has been proposed.

The forward-node set and, consequently, the self-pruning problems can essentially be viewed as the task of solving the NP-hard "Minimum Connected Dominating Set" (MCDS) problem [159]. Several studies ([172], [173], [149], [150], [151]) have attempted to tackle the MCDS problem by constructing a communication backbone prior to the broadcast initiation. These schemes can sometimes dramatically reduce the number of transmissions. For instance, Funke's algorithm in [150] provably guarantees a 6.94-approximation ratio to the size of the MCDS. Nevertheless, as shown in [174], such backbone schemes do not tolerate well frequent network topological changes. For volatile communication environments, an approach to dynamically construct CDS is a better alternative. Works such as [175], [154], and [136] offer initial solutions. However, [154] relies on positional information such as GPS, and GPS is not always feasible. The algorithm in [175] does not scale well for higher density networks. Finally, the TSS scheme described above and also in [155] is significantly more efficient in reducing the number of retransmissions (i.e. finding better approximation to the MCDS) than the CCS algorithm in [136].

Bounds on the size of $|MCDS|$ in unit disk graphs (UDG) have been studied extensively in the technical literature, as well. Typically these bounds are given via ratio between the cardinalities of the Maximal Independent Set (MIS) and MCDS. Let $\mu = |MCDS|$ and $\chi = |MIS|$. Wu et al. demonstrate that $\chi \leq 4\mu + 1$ [176]. Further improved bound is derived in [177], where $\chi \leq 3.43\mu + 4.82$. Finally, a tight (non-improvable) bound is derived by Vahdatpour et al.: ([178])

$\chi \leq 3\mu + 3$.

A number of studies have attended to the theoretical complexity bounds of broadcast and related information dissemination mechanisms for topology (not area) bounded networks. For instance, [179] and [180] provide such complexity bounds on gossiping, broadcast, and rumor spreading in such networks. Also, influential works by Peleg et al. in the spirit of [181] demonstrate important complexity lower bounds on broadcast in radius-2 radio networks showing that the broadcast procedure requires $\Omega(log2n)$ transmissions.

4.3 Advantages and Limitations of the TSS scheme

The Time Sequence Scheme for broadcasting in wireless networks described above is based on finding a distributed approximation of the wireless network MCDS. Through simulations and based on two metrics - the transmission complexity and the delay - TSS performance has been compared to other leading broadcasting schemes. The TSS scheme outperforms all other schemes with respect to the number of broadcast message transmissions, without requiring additional equipment, such as GPS. Furthermore, this performance is achieved with bounded latency, and is independent of network density.

TSS improves on the performance of CCS and RBS with respect to number of transmissions and network coverage, while achieving comparable or better delay than the two algorithms. TSS also possesses network partitioning immunity and outperforms the other schemes with respect to network coverage in all mobility models simulated, achieving almost full coverage. This feature makes TSS one of the few alternatives to flooding in the context of mobile networks.

TSS is among the first broadcast algorithms, satisfying the efficient broadcast desiderata (1)-(5) of section 4.1.1, and applying the TSS scheme in various network scenarios such as deployment of UAVs and sensor networks is a practical option.

However, note that with the current implementation of the TSS broadcast algorithm, its deployment in extremely dense and dynamic networks may not be practical. The number of control packets exchanged to compute *RC* in the latter settings may be rather large leading to increased delay. An alternate strategy of estimating *RC* that reduces the number of control packets in such cases may be needed. For instance, instead of replying to a residual coverage request message 100% of the time, only a fraction of the nodes may reply (depending on the network density; the higher the density the smaller the fraction). Thus, a node may only have an estimate of the *RC*. This approach has not been implemented yet, and a study of such alternate *RC* computation is left for future work. The TSS performance though is robust both in mobile and static networks containing up to a few hundred nodes.

CHAPTER 5

FUTURE WORK AND CONCLUSION

5.1 Future Work: Extensions

The latter chapters laid out a preliminary set of perspectives towards designing data level aware algorithms for the structural and procedural wireless network levels. For instance, the Encoded Sensing (ES) scheme of chapter 2 accounted for the statistical properties of the data source (e.g. spatial correlation in sensor nodes' measurement, common data due to the broadcast nature of the wireless transmissions, etc.) to coordinate and encode the transmissions of specifically placed representative groups of nodes to reduce wireless signals sent to the sink. Thus, ES is "data-aware" while shaping the structural level of the network. Chapter 3 discussed further the optimization of the structural level via the placement of additional relay nodes and determining the set of all pairs, unicast, single path routes, so that the communication cost (i.e. retransmissions overhead) in the network is minimized. Given the placement of the representative groups of nodes, relay nodes may be positioned to augment the links on heavily loaded paths connecting these groups. Finally, chapter 4 described the Time Sequence Scheme broadcast protocol that minimizes the number of required transmissions in the network, so that all nodes are covered by a single-source broadcast message. Hence, given the data-aware structural level designed in 2 and 3, the TSS could be used as a solution to propagate network wide updates across all different representative groups of nodes and relay nodes.

While the algorithms presented within each individual network level provide significant improvement and new perspectives on wireless networks systems, the interactions across the data, structural and procedural levels need to be studies further. Additionally, a number of technical questions and challenges regarding the design of algorithms at each level remain to be addressed. A brief overview on some of those is given below.

5.1.1 Topology Control for Clusters of Collaborative Nodes

In chapter 3, the placement of relay nodes assumed that each relay transmission is independent of the transmissions of other relay nodes. However, in practice, relay nodes may utilize any of the *cooperative* transmission schemes discussed in section 2.6 including ES, to synchronize their transmissions and achieve better SINR on cooperative links. To achieve that, instead of placing individual relay nodes utilizing the link model in section 3.2.1, subsets of the available relay nodes may be clustered together to transmit collaboratively according to cooperative transmission models similar, for instance, to the ones described in [86], [182], and [85] among others. The technical challenge here is to *determine the number of nodes within each relay cluster and determine the position of each relay cluster and the nodes within it, so that the the network goodput is optimized*. This question has not been addressed yet in the technical literature and is open for future work.

The benefits of such setup would allow the extension of the encoded sensing scheme of chapter 3 to multi-hop network scenarios, while minimizing the number of overhead retransmissions due to weak transmission links. As a result, for example, the recent energy-efficient wireless ad-hoc network protocol relying on cooperative transmissions suggested in [86] could be simplified with

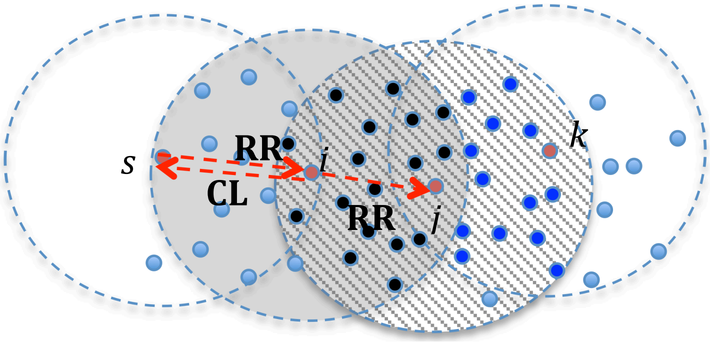


Figure 5.1: CMF: the black nodes form G_i after receiving REC_i and sending GR_i from and to i respectively. Next, i sends a CL packet to s and RR packet to j. j forms G_j (dark blue nodes). Per ES, K_i^* nodes in G_i transmit at hop i. Up to $\binom{18}{9} = 48620$ (i.e. 16-bit) messages can be encoded by G_i when $c_x = |A_x| = 9$. Notice that the CF packet from [86] is now not required. The challenge is to position relay nodes in clusters within the network, so that the collaborative transmissions of each cluster reduce the number overhead retransmissions (i.e. network communication cost per equation (3.9) derived in the context of cooperative transmissions).

ES, via relaxing assumptions on the physical layer. Extending ES application to multi-hop network scenarios, would allow ES to reduce the control overhead of the cooperative protocol in [86] at the procedural network level.

The cooperative message forwarding protocol (CMF) introduced in [86] consists of two phases: *routing* and *recruiting-and-transmitting*. In the routing phase a multi-hop path L is established between a source and destination node, employing ad-hoc routing such as the Zone Routing Protocol.

Only the details relevant to ES of the *recruiting-and-transmitting* (RT) phase of CMF are described briefly here and illustrated in fig. 5.1. During the RT phase, each node in L recruits a group of nodes in its neighborhood dynamically, as the message x is propagated on the path L. This is done via a series of control and synchronization message exchanges between neighboring nodes in L. Suppose nodes s, i, j, and k in that order are nodes in the path L. When the message x reaches the hop at node s, s sends a *request-to-recruit* (*RR*) packet to i. Node i forms a cluster of nodes in i's neighborhood that will cooperatively transmit the message to the next hop j. Upon receiving *RR*, i sends a *recruit* (*REC*) packet to its neighbors including, among other control information, the ID of the next hop, j, in L. The nodes that are both neighbors of i and j form a cluster G_i and reply to i with a *grant request* (*GR*) packet. i is now the clusterhead of G_i and in charge of coordinating the cooperative transmission. The nodes in G_i are ready to receive x and then cooperatively transmit x to the next hop. i sends s a *clear-to-send* (*CL*) packet. In addition i is required to send a *confirm* (*CF*) packet to the nodes in G_i that contains the *waiting-time-to-send* and their transmission power level. i then sends *RR* packet to j and the operation repeats. It is assumed that the cooperating nodes synchronize each other's transmitted signals at the receiver on a bit level in order to claim the benefits of cooperative transmission. Notice that this implies very precise clock synchronization across cooperating nodes. In [86], groups of sizes up to 5 nodes are discussed, since synchronizing more than 5 nodes dynamically while the message is propagated on the path L is rather challenging and incurs communication overhead. Even in this case, CMF achieves substantial energy savings over non-cooperative message forwarding schemes.

At this point one can observe that 1) encoded sensing can be naturally incorporated in CMF, replacing cooperative transmissions at the physical layer, and extending the application of ES into multi-hop network scenarios; 2) incorporating ES, relaxes CMFs synchronization assumptions and reduces CMFs control messages communication overhead as follows. On the path L, form G_i with clusterhead i exactly as per CMF. Upon receiving message x, the nodes in G_i run through the assignment and encoding stages of ES as described in section 2.1.2. Per CMF, i sends RR packet to k and after a timeout a CL packet is received from k in the cluster G_i. Once the CL packet is received in G_i, the nodes in A_x of G_i execute the transmission stage of ES. Per CMF, k has already formed a receiving cluster G_k that receives the transmissions of the nodes in G_i. The nodes in G_k perform the decoding stage of ES to obtain x; they in turn run through the ES assignment and encoding stages. That way, similarly to the original CMF, the message is forwarded hop-by-hop on the path L. However, here, the control message CF along with the bit-level synchronization on the physical layer are not needed. As demonstrated in chapter 2 ES can be as energy efficient as distributed transmit beamforming (e.g. [34]) with number of beamforming nodes up to 5.

As noted, algorithms for placing clusters of relay nodes that could utilize ES and the other cooperative transmissions schemes, as described above, in order to minimize retransmissions in wireless networks are not known, and further work in that direction could be impactful.

5.1.2 Designing a Hybrid Structural-Procedural Level

The schemes described in the preceding chapters could be further improved by examining the trade-offs between designing "data-aware" structural levels and "procedure-aware" structural levels. For instance, in chapter 2 and 3, the suggested schemes determined network structure via topology control assuming respectively certain properties of the data sources and unicast routing in the network. However, if the applications running over the network require primarily multi-cast and broadcast data dissemination solutions, the placement both of relay nodes and representative groups of nodes may be suboptimal. To provide flexibility in that regard, a different multi-level approach could be considered. For example, optimizing the structural network level for broadcast *procedures* would lead to a set of algorithms for topology control resulting in topologies of type Γ_p perhaps similar to the one described in Appendix D.1 if the network is dense. Optimizing the structural network level considering the *data* level and using ES, while placing relay nodes according to chapter 3, would lead to topologies Γ_d. Over-simplifying the problem in the interest of presentation, consider a network utilizing broadcast for $\alpha\%$ of the transmitted data and unicast for $(1-\alpha)\%$. A hybrid policy for wireless network design could "weight" the output of algorithms producing topologies Γ_p and Γ_d according respectively to the proportion $\alpha\%$ to derive a "hybrid" topology Γ_h that optimizes the number of overhead transmissions in the network considering both the data and procedural network levels.

5.1.3 Load Balancing, Load Estimation and Relay Placement

Chapter 3 considered the optimal placement of relay nodes to optimize communication cost in the resulting wireless network. One of the implicit assumptions in the problem formulation presented in the chapter is that traffic loads, distribution of source-destination pairs, and traffic demand matrices may be arbitrary and are unknown in advance. In various scenarios, a number of routing algorithms could be utilized that balance the traffic across the network links (e.g. [78], [79], [80]). That is, in large scale networks with multiple redundant paths connecting source-destination pairs and relatively uniform distribution of source-destination pairs, a routing algorithm may be selected that achieves relatively uniform traffic across all links in the network, independent on the particular positions of source-destination pairs and relay nodes. In fact, under these assumptions, a single, minimum-weight path routing algorithm often load balances the traffic offered to network nodes as studied in [183]. That is given the cumulative traffic demand one could estimate the load on each link in a load balanced network down to a certain level of uncertainty. Furthermore, given a shortest path routing algorithm and network nodes' positions one could again estimate the traffic loads on network links without explicitly computing the routing paths in the network ([183]).

Let the traffic load on link ij be T_{ij} where i and j could be any network nodes. T_{ij} could be modeled as a random variable: a sum of (deterministic) estimate E_{ij} for the traffic load on link ij and a noise term Z_{ij} representing the uncertainty in the traffic load estimate.

$$T_{ij} = E_{ij} + Z_{ij} \tag{5.1}$$

Again, notice that in this setting, we do not consider the explicit computation of different possible load-balance routing patterns. Instead, we encode the variation across the different deterministic load balanced routing patterns in the uncertainty of the traffic load estimate. One then could utilize various methods from stochastic optimization such as Sequential Kringing Method (SKM [184]) and Simultaneous Perturbation Stochastic Approximation (SPSA [185]) to optimize a cost function similar to the on in eq. (3.16). Under certain assumptions on the properties of the cost function in (3.16) (smoothness and convexity), SPSA and SKM guarantee convergence to the global optimum w.h.p.. A different approach in this probabilistic load-balanced setting, may utilize a number of works on chance (a.k.a. probabilistic) constrained programming (e.g. [186]

and [187]), where (5.1) places probabilistic constraints on the traffic loads that are feasible. In either case, the reformulation of the optimization problem may not be trivial, but if done properly would yield a practical algorithm with analytically optimal solution w.h.p..

5.2 Future Work: Research Directions

Aside from the potential extensions discussed in the above section which could improve on the work presented in this thesis, a few broader avenues for research can also be identified. The sections below briefly outline two of those.

5.2.1 Sparse DSSS for Varying Capacity CDMA Systems

Section 2.3 suggests and studies a novel DSSS receiver design inspired by the signal sparsity inherent for ES operation. The sparse DSSS receiver design may have potential applications in CDMA systems (not limited to sensor networks) where the set of active users at any given time is much smaller then system's capacity. Such systems could comprise cellular networks in rural areas, for instance. Even in these cases the number of active users in the system would not be exactly constant but would vary. The system is not exactly K-sparse, where K is the number of active users, at any given instance in time. This variation in sparsity needs to be handled without modification to the hardware at the receiver's matched filters, while preserving the reduced circuitry complexity made possible via compressive sensing. Furthermore, even rural systems may suffer peak utilization spikes. During these spikes the received signal is not sparse and the receiver needs full circuitry complexity (i.e. M matched filters handling each of the M active users under 100% utilization).

Hence, despite being of potential practical and theoretical interest, the general sparse DSSS receiver hardware design's use cases may be constrained.

A more flexible solution is possible by considering a dual problem of the one studied in section 2.3. Instead of modifying the reference PN sequence signature waveform at the receiver to be the linear combination given in eq. (2.14), one could modify the PN sequence signature waveforms at the source nodes using equation similar to (2.14). The DSSS receiver hardware would consist of one matched filter for each user in the system as in a regular DSSS system (the reference PN sequence waveform at each matched filter may still require modification). The PN sequence at each user is changed with time, depending on the number of active users in the system. In this case, the reduced cost in the system is not in terms of hardware complexity, but rather in terms of energy consumption for each of the active users. If the system is K-sparse the lengths of the spreading sequence utilized by each user would be on the order of a logarithm of the original PN sequences length. As the system utilization grows, the spreading sequence at each user also grows in length. At peak utilization, the spreading sequence converges to the original PN sequence.

Such a system could significantly reduce devices energy consumption in rural area CDMA systems, while maintaining overall system peak capacity. Analytical, simulation and experimental studies of the parameters of such sparse, variable length spreading sequences and correspondingly the design of the matched filters at the receiver may have significant research and practical impacts.

5.2.2 Underwater Networks

Throughout this work, we have considered wireless radio signals assuming terrestrial signal propagation. In the context of underwater sensor networks, where communication occurs via acoustic wireless links, a number of questions regarding the structural network level and its interaction with the data level remain open.

Acoustic signal's path loss function, multipath effects and link time variability are rather different than the ones of terrestrial radio systems. The relay placement problem investigated in chapter 3 takes different form. To the best of the author's knowledge the relay placement problem in the setting of underwater networks has not been investigated well. One of the few available studies of the problem has been published only recently ([188]). While the authors analyze the effect of acoustic wave frequency and separation distance between relay nodes on network performance, their approach does not take into account a number of factors such as routing scheme, effects of multipath, etc..

Encoded Sensing could be suitable for low rate, underwater acoustic sensor networks. However, the analysis of the scheme in this scenario would differ. The DSSS PN sequence acquisition techniques are different from the ones utilized in terrestrial systems ([189]). Although, certain acquisition performance metrics have been studied for specific acquisition schemes, an information theoretic acquisition-based capacity of these systems has not been derived and is still unknown.

5.3 Conclusion

Wireless network performance and lifetime is directly related to the number of transmission signals, ceteris paribus, sent by network nodes. In this thesis, the wireless network is decomposed in three levels: the data, structural and procedural levels. We have considered a distinct communication scheme coupled with each level, optimizing the properties of the respective level, so that the number of (re)transmissions (i.e. communication cost) is minimized. Fig. 5.2 illustrates the general overview of the presented work. At the interface between the data and structural level, we have introduced the Encoded Sensing (ES) paradigm that allows collaborative division of responsibility across network nodes with access to common or approximately common data. Chapter 2 discusses the implementation of ES and demonstrates that the scheme generates at least a factor of two less transmissions than state-of-the-art conventional non-collaborative transmission. ES achieves energy efficiency comparable to distributed beam-forming (DBF) techniques. However, DBF relies on very tight synchronization across source nodes that is hard to scale in practice, whereas ES only requires standard DSSS physical layer. A novel sparse DSSS receiver design is proposed and analyzed based on the inherent sparsity of the ES signal received at the sink node. The sparse DSSS receiver has substantially simplified circuitry and may lower the cost of DSSS receivers in applicable scenarios, independent of ES.

In chapter 3, the structural network level is optimized by the placement of relay nodes minimizing network communication cost. Although the relay placement problem seems well studied in the technical literature, we identify a number of inherent inefficiency in the problem's models currently utilized by

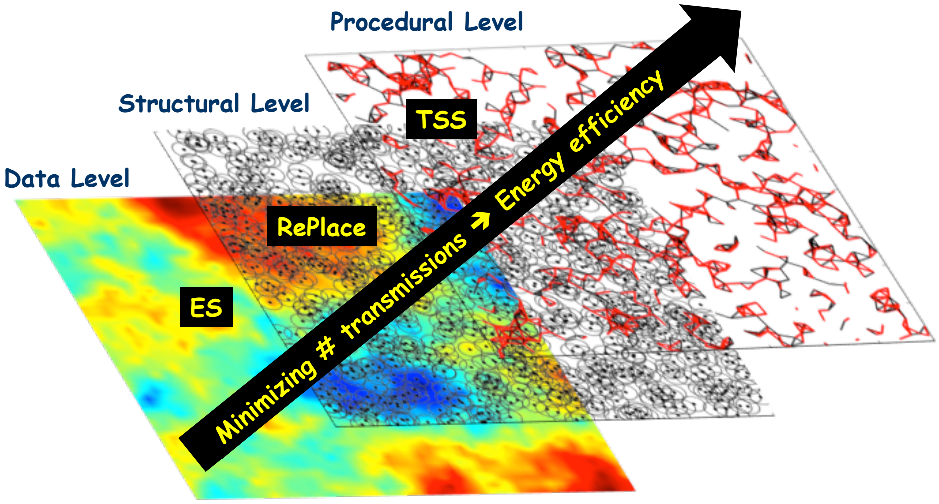


Figure 5.2: Network levels and respective schemes described in the thesis

network researchers and practitioners. We highlight the importance of the feedback loop between traffic loads/routing and relay nodes positions and suggest a brute force optimal algorithm. A heuristic (the RePlace algorithm) designed to maximize the communication cost improvement resulting from the placement of relay nodes is described to eschew the brute force algorithm exponential complexity. RePLace performs almost identically to the optimal solution in small networks. In larger networks, RePlace eliminates overhead network transmissions almost completely, as the number of relay nodes increases. Simultaneously, we show that the average packet delay in the resulting relay network decreases significantly. In comparison to other relay placement algorithms with the goal of minimizing communication cost from the technical literature, RePlace performs better both in terms of communication cost and delay reduction.

Given the optimized structural level, accounting both for the data level characteristics (e.g. correlation of measurements in WSN) and relay placement, we consider the optimization of important primitive at the network procedural level. In chapter 4, we introduce the Time Sequence Scheme (TSS) broadcast protocol designed to compute an approximation of the Minimal Connected Dominating Set (MCDS) in wireless network graphs. TSS is implemented in the JiST/SWANS simulator and compared against a multitude of the best established and very recent broadcast algorithms. TSS reduces the number of broadcast nodes, or equivalently the size of the MCDS, to a fraction of network nodes substantially less than competitive schemes. Furthermore, TSS remains robust within highly dynamic networks, where temporary disconnections during algorithm's operation are possible.

As a result of our study, at different network levels, we confirm that carefully reducing the number of (re)transmissions in a wireless networks indeed improves not only network throughput, but also in most cases improves quality of service. At the data level we observed that encoded sensing does not sacrifice accuracy of estimate, while sending fewer signals from sensor nodes to the sink. Optimizing the structural and procedural level to reduce the number of overhead and redundant transmissions in the network, invariably decreased average packet latency.

The methods presented in this work pose a number of questions for further research that may benefit practical deployment of future sensor and wireless networks. The preceding sections of the current chapter outline some of the possible extensions of our techniques to improve their robustness and performance. Furthermore, we consider new, broader directions for future work with potentially high research and industrial impact.

APPENDIX A

ACQUISITION CAPACITY OF DSSS SYSTEMS

The signal acquisition stage is inherent part for any DSSS-based system (e.g. [93], chapter 3). During this stage, the delay of node *j*'s signal (and similarly of the remaining transmitting nodes) is estimated, so that its transmission can be properly matched and despread at the receiver (e.g. see [94]).

In the case of asynchronous acquisition system a lower bound on the capacity of the system can be obtained, so that the probability of bit error vanishes to 0. This is the probability that a matched filter would erroneously determine that its reference waveform matches one of the other signature waveforms in the received signal. Although the detailed analysis is omitted here, it follows directly the derivation of the asynchronous system acquisition capacity lower bound of Appendix B in [94]. As shown in [94], the acquisition capacity determines the capacity of an asynchronous DSSS system. The lower bound on the system capacity is given by

$$N < \frac{\alpha^2 \left(F log F\right)}{2\bar{P}}$$

where F is the length of the matched filter acquisition window length (a parameter of the sink's receiver design determining its circuitry complexity) and $\bar{P}$ is given by

$$\bar{P} = \frac{1}{N}\sum_{i=1}^{N} P_i$$

If all the received powers of nodes' transmissions are approximately the same then, $\bar{P} \rightarrow 1$ and the capacity of the system is equal to

$$N < \frac{1}{2}\alpha^2 \left(F log F\right) \tag{A.1}$$

The latter assumption is feasible when modeling the transmissions of nodes within a single group G_i. The nodes within the same representative group (see Definition (2.1)) are clustered spatially so that the measurements of nodes within the same representative group are highly correlated. Thus, the near-far effect in the system is minimal. The different representative groups in the system could transmit in different timeslots to avoid interference at the sink.

APPENDIX B

DISTORTION OF MEASUREMENT PER ES AND NCD

The estimate of the source **S** at the sink produced utilizing either ES or NCD is distorted since 1) only information regarding the measurements of k representative out of the N nodes in the network is taken into account; 2) there is channel noise and sensing imprecision noise present. The CC-MAC algorithm in [21] utilized for representative node selection both by NCD and ES ensures that the number of selected k nodes is *minimized* so that the resulting distortion $D(k)$ is within a quality of service constraint D_{QoS}. We demonstrate that ES achieves similar level of distortion. For both schemes the distortion is given by standard Minimum Square Error (MSE) metric:

$$D(k) = E\left[(\mathbf{S} - \mathbf{S}')\right] = E\left[\mathbf{S}^2\right] - 2E\left[\mathbf{SS}'\right] + E\left[\mathbf{S}'^2\right] \tag{B.1}$$

where **S** is the value of the point source; **S′** is the estimate of the point source at the sink given the k reports. Suppose the channel noise is AWGN so that $Z_j \sim N(0, \sigma_Z^2)$.

For a non-cooperative transmission scheme such as NCD, the measurement X_j of a representative node j is transmitted most efficiently using uncoded transmission (ref. [190]) subject to power constraint P per node, per measurement. The received signal at the sink is

$$Y_j = \sqrt{\frac{P}{\left(\sigma_Z^2 + \sigma_W^2\right)}} \cdot \left(S_j + W_j\right) + Z_j$$

The optimal decoder at the sink is given by the standard MMSE estimator [190]:

$$S'_j = Y_j \cdot \left(\frac{E\left[S_j Y_j\right]}{E\left[Y_j\right]}\right)$$

and therefore

$$\mathbf{S}' = \frac{1}{k}\sum_{j=1}^{k} S'_j$$

Evaluating (B.1) term by term and simplifying yields

$$D_{NCD}(k) = D(k,P) = \sigma_S^2 - \sigma_S^4 \left[\left(\sigma_S^2 + \sigma_W^2\right)\left(1 + \frac{\sigma_Z^2}{P}\right)\right]^{-1} \times$$

$$\left[\frac{1}{k}\left(2\sum_{i=1}^{k}\rho_{S,i} - 1\right) - \frac{\sigma_S^2}{k^2}\sum_{i=1}^{k}\sum_{j=1,j\neq i}^{k}\frac{\rho_{i,j}}{\sigma_S^2 + \sigma_W^2} - \frac{\left(1 - \frac{1}{k}\right)\sigma_Z^2}{P}\right] \quad \text{(B.2)}$$

Per ES, $|A_x| = K$ nodes transmitting for each of the k representative groups. The nodes transmit in a multi-access interference limited system utilizing DSSS. The channel noise is negligible in comparison to nodes' mutual interference. However, since the system operates at acquisition capacity, the probability of transmission error due to interference is close to 0 (see Appendix A and [94]) and consequently does not incur distortion of measurement. Here, we have assumed the range of values a phenomenon can take is split in intervals of length $\epsilon = 2\beta\sigma_W$, $\beta > 2$. Hence, per each group G_j, the estimation of measurement at the sink is $S'_j = S_j + 2\beta\sigma_W$.

As before $\mathbf{S}' = \frac{1}{k}\sum_{j=1}^{k} S'_j$ and plugging in (B.1) yields:

$$D_{ES}(k) = \sigma_S^2\left[1 - \frac{1}{k}\left(2\sum_{i=1}^{k}\rho_{S,i} - 1\right) + \frac{1}{k^2}\sum_{i=1}^{k}\sum_{j=1,j\neq i}^{k}\rho_{i,j} + \left(\frac{2\beta\sigma_W}{\sigma_S}\right)^2\right] \quad \text{(B.3)}$$

Aside from the term $\left(\frac{2\beta\sigma_W}{\sigma_S}\right)^2$, as shown in fig. 2.7, ES achieves similar distortion to NCD as the number of representative groups/nodes increases and $\sigma_W < \sigma_S$. The latter condition is satisfied as the sensor instruments on nodes become more accurate.

APPENDIX C

CONVEXITY OF THE RPFT PROBLEM

From Theorem (3.1), one only needs to show that the function $g(z)$ is convex and non-decreasing for $z \in (0, rd_0)$, where $r > R$. For clarity of presentation, assume $d_0 = 1$.

To analyze the above claim, consider the function

$$g(z) = \left[1 - Q\left\{\sqrt{2\gamma(z)}\right\}\right]^{-b} \tag{C.1}$$

where $z \in \mathbb{R}^+$. Substituting γ from (3.2), $g(z)$ now equals

$$g(z) = \left[1 - Q\left\{\sqrt{2\left(P_t - a(1) - 10\alpha\log_{10}(z) - \eta(0,\sigma) - \chi(0,\sigma_1)\right)}\right\}\right]^{-b} \tag{C.2}$$

Let $A = P_t - a(1) - \eta(0,\sigma) - \chi(0,\sigma_1)$. A is a Gaussian r.v., however, at present A is treated a constant w.r.t. to z. Then

$$g(z) = \left[1 - Q\left\{\sqrt{2\left[A - 10\alpha\log_{10}(z)\right]}\right\}\right]^{-b} = \left[\frac{1}{2} + \frac{1}{2}erf\left(\frac{\sqrt{2\cdot 10^{\left(A - 10\alpha\log_{10}(z)\right)/10}}}{\sqrt{2}}\right)\right]^{-b}$$

where the $Q(\bullet)$ function was converted to the error function $erf(\bullet)$ and the SNR expression (2.2) was converted from dB to linear scale. Let $B = 10A/20$ and $a = \alpha/2$. We obtain

$$g(z) = \left(\frac{1}{2}\right)^{-b}\left[1 + erf\left(Bz^{-a}\right)\right]^{-b} \tag{C.3}$$

The multiplicative constant $\left(\frac{1}{2}\right)^{-b}$ does not affect the convexity of $g(z)$. Hence, we only consider

$$g_1(z) = \left[1 + erf\left(Bz^{-a}\right)\right]^{-b}$$

If $g_1(z)$ is convex, so is $g(z)$.

$$\frac{dg_1(z)}{dz} = \frac{2abB}{\sqrt{\pi}} \cdot \frac{e^{-Bz^{-2a}}\left[1 + erf\left(Bz^{-a}\right)\right]^{-b-1}}{x^{a+1}} \tag{C.4}$$

Note that on $\mathbb{R}^+$, $g_1'(z)$ is non-negative for any z. Hence, $g_1(z)$ and respectively $g(z)$ is non-decreasing.

The constant $\frac{2abB}{\sqrt{\pi}}$ is positive and can be ignored. Using the product and quotient rules

$$\frac{d^2 g_2(z)}{dz^2} = a\left[1 + erf\left(Bz^{-a}\right)\right]^{-b-1} e^{-B^2 z^{-2a}} \cdot \left\{ \frac{2(b+1)Be^{-B^2 z^{-2a}}}{\left[1 + erf\left(Bz^{-a}\right)\right]\sqrt{\pi}} + \frac{2B^2}{z^a} - \frac{z^a(a+1)}{a} \right\}$$

If $g_2''(z)$ is non-negative, $g(z)$ is convex. Notice that since $a > 0, b > 0$, and $B > 0$

$$a\left[1 + erf\left(Bz^{-a}\right)\right]^{-b-1} e^{-B^2 z^{-2a}} > 0$$

for all z. Hence, we are only interested in the inequality

$$\frac{2(b+1)Be^{-B^2 z^{-2a}}}{\left[1 + erf\left(Bz^{-a}\right)\right]\sqrt{\pi}} + \frac{2B^2}{z^a} - \frac{z^a(a+1)}{a} > 0 \tag{C.5}$$

In this form (A.5) does not have analytical solution expressed in simple functions.

However, one can still reason about the convexity of the function $g(z)$. First, simplify (A.5) further by noting that $\left[1 + erf\left(Bz^{-a}\right)\right] < 2$ for any z, since Bz^{-a} is positive. Therefore (A.5) is valid if the inequality

$$\frac{(b+1)Be^{-B^2 z^{-2a}}}{\sqrt{\pi}} + \frac{2B^2}{z^a} - \frac{z^a(a+1)}{a} > 0 \tag{C.6}$$

is satisfied. Also, observe that $B^2 z^{-2a} < 0.15$. Hence, from the Taylor series expansion of $e^{(\cdot)}$

$$e^{-B^2 z^{-2a}} \approx 1 - B^2 z^{-2a}$$

This approximation is valid as z (or equivalently, the distance between two network nodes) increases, since $B = 10A/20$ and $a = \alpha/2$. Typically in wireless networks models $\alpha = 3$ and B is small in low SNR regimes. Using this approximation and slightly rearranging the resulting inequality from (A.6) we obtain

$$\frac{2\sqrt{\pi}B(b+1)}{z^a} - \frac{B^2(b+1)}{z^{3a}} - \frac{\sqrt{\pi}(a+1)}{aB} > 0 \tag{C.7}$$

Noting that $z^a > 0$, substituting$\alpha = 3, a = \alpha/2$ and letting

$$\zeta_1 = \frac{\sqrt{\pi}(a+1)}{aB}, \zeta_2 = 2\sqrt{\pi}B(b+1), \zeta_3 = B^2(b+1) \text{ and } y = z^{3/2}$$

the cubic inequality

$$\zeta_1 y^3 - \zeta_2 y^2 + \zeta_3 < 0 \tag{C.8}$$

is obtained. The solution intervals of (C.8), after reverse substitution, are equivalent to the intervals where $g(z)$ is convex in low SNR regimes.

The solutions of (C.8) can be found explicitly, however the roots are complicated. Instead, one can gain intuition about the asymptotic behavior of the function $g(z)$ by observing its first derivative as z increases.

Recall that

$$\frac{dg_1(z)}{dz} = \frac{2abB}{\sqrt{\pi}} \cdot \frac{e^{-Bz^{-2a}}\left[1 + erf\left(Bz^{-a}\right)\right]^{-b-1}}{x^{a+1}} \tag{C.9}$$

Notice that as z increases $\left[1 + erf\left(Bz^{-a}\right)\right]^{-b-1} \approx 1$.

Then, asymptotically

$$\frac{dg_1(z)}{dz} = \frac{2abB}{\sqrt{\pi}} \cdot \frac{1 - B^2 z^{-2a}}{z^{a+1}} \to 0 \tag{C.10}$$

Then, notice that $g_2''(z)$ 0. Thus, the function $g(z)$ is convex in this asymptotic regime. (More precisely the function $g(z)$ is non-strictly convex.) We notice that intuitively this is correct since the $Q(\cdot)$, analogously $erf(\cdot)$, and the packet error rate functions converge to constants, as z increases. I.e., the received power is very low at a node that is large distance away from a transmitting node.

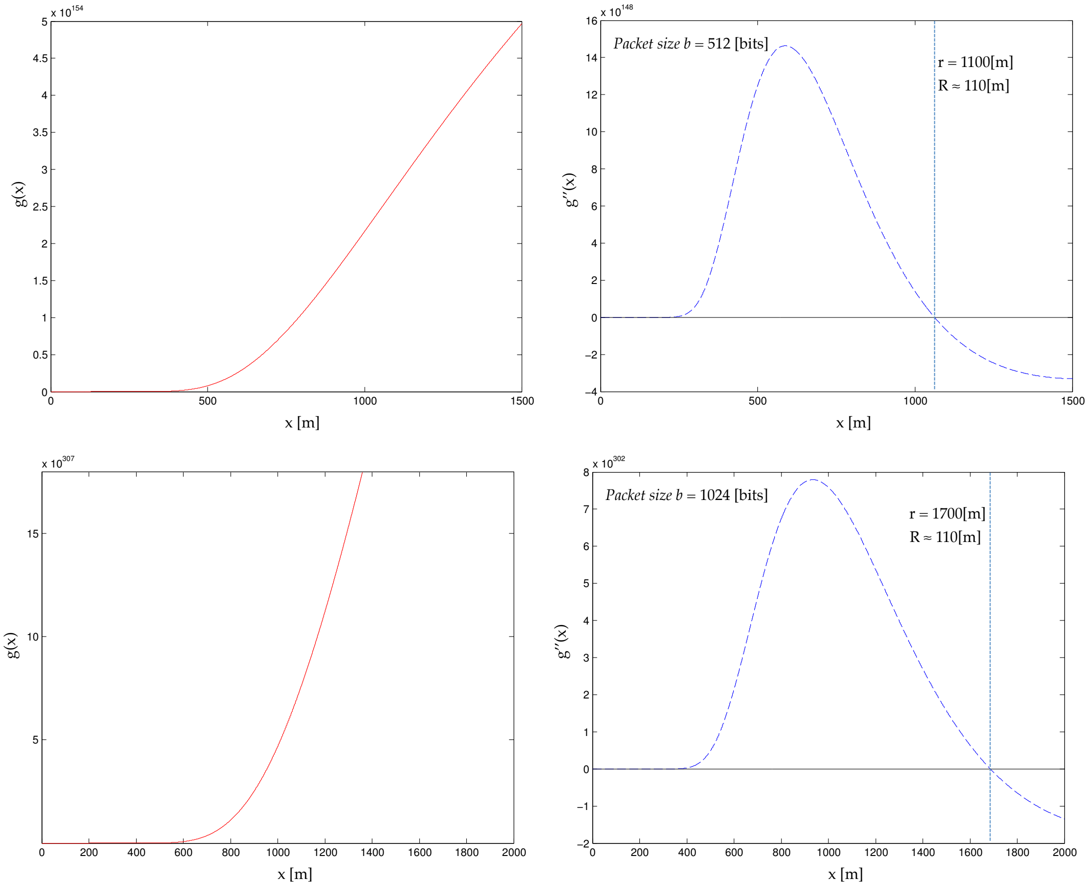


Figure C.1: $g(x)$ and $g''(x)$ for different packet sizes and fixed transmit power of 10[W]. Notice that the convexity interval $(0, r)$ increases as b grows. The transmission range, $R \approx 110$[m] is less than r for various values of BER and SNR.

This regime occurs if transmission links between nodes are long, the packet error rate is very high, and links' capacities are low. We note that as z decreases the first two terms in (C.8) increase and the third term decreases. Qualitatively, as z decreases the function becomes strictly convex.

Let $z \in (0, r)$ and respectively $z \in (0, rd_0)$ be intervals where the function is convex. Figure C.1 demonstrates that (C.8) is indeed satisfied for large values of r depending on the transmit power and the packet size. The larger the transmit power and the packet size, the larger r. For instance, if the transmit power is

10[W] and the packet size is 512 bits the function $g(z)$ is convex within a range of $z = rd_0 = 1100[m]$, for packet size of 1024 bits; $g(z)$ is convex within a range of $z = rd_0 = 1700[m]$. $d_0 = 1[m]$ throughout. Also, to increase the transmission range, the transmit power is increased, which in turn increases the convexity interval $(0, r)$.

APPENDIX D

BOUNDS ON WIRELESS BROADCAST

This section provides upper and lower asymptotic bounds on the number of retransmissions in a 2D wireless ad hoc network. The bounds are demonstrated based on geometrical arguments (given the system model described in section 4.1.2). The bounds are general and hold for any broadcast algorithm over UDG.

Typically the bounds on the size of MCDS for UDG are presented in terms of an approximation ratio between the cardinality of the Maximal Independent Set (MIS) and the cardinality of the MCDS (section 4.2) provides more details and references). In contrast, the bounds shown below are simple, useful for the purposes of algorithms benchmarks, and are characterized by closed form expressions depending only on the size of the area enclosing the network and nodes' transmission radius, R. This is useful for high density networks since computing the exact size of MCDS is NP-hard problem ([139]), which of course is intractable for large number of nodes. The running of algorithms (be they efficient) finding constant factor approximations either to MIS or MCDS for every graph instance may neither be feasible nor required.

Let the minimum number of nodes transmitting a broadcast message (equivalently number of transmissions) be β_{min} the number of nodes in the network be $N = |S|$, the area enclosing the network be $\mathcal{A}$, its size be a constant $|\mathcal{A}|$, the set of all points in $\mathcal{A}$ covered by transmission of node i be $\diamond_i$, and $\mu = |MCDS|$. As noted in section 4.1.3, $\beta_{min} = \mu$.

Let the area $\mathcal{A}$ be rectangular. As $N \to \infty$, with high probability one can find and pick a node at any desired point in $\mathcal{A}$. Given graph topology $\Lambda(S)$, the goal is to pick a set of nodes X positioned so that

i $\bigcup_{i \in X} \diamond_i$ covers $\mathcal{A}$ entirely;

ii $\bigcap_{i \in X} \diamond_i$ is minimized (i.e. the overlap between transmissions is minimal).

iii for any two nodes i and j in X, i and j can communicate: $\exists$ a path of nodes entirely in X connecting i and j.

Let $\Phi(\Lambda(S))$ be the conjunction of i), ii), and iii) given topology $\Lambda(S)$.

D.1 Lower Bound on the Number of Transmissions

As $N \to \infty$, $\Phi(\Lambda(S))$ yields a recast of the MCDS problem with solution the set X in the context of a bounded area $\mathcal{A}$ and UDGs. Namely, $|X| = \beta_{min}$.

Now, let $\diamond_i = D_i$ where D_i are identical disks with radius R (i.e. all nodes in the network have the same transmission range). Under the constraint of $\Phi(\Lambda(S))$, let $|X_D| = |X| = \beta_{min}$. It is easy to check that utilizing circular disks D_i to satisfy i) of $\Phi(\Lambda(S))$ would introduce substantial overlaps between the disks covering area $\mathcal{A}$. Instead, suppose each of the N disks D_i is approximated with a hexagon H_i with side R and centered at i. That is, under the constraint of $\Phi(\Lambda(S))$, let $\diamond_i = H_i$ and let $|X_H| = |X|$ in this case. The plane is readily tessellated by hexagons thus avoiding the overlaps needed to satisfy i). In essence, $\beta_{min} = |X_D| > |X_H|$ as long as a construction is provided so that ii) and iii) are also satisfied, given $\diamond_i = H_i$.

Consider the Linear Hexagon Packing (LHP) construction as described in fig. D.1. A node from S is picked to be in X if it is at a centroid of a hexagon in the

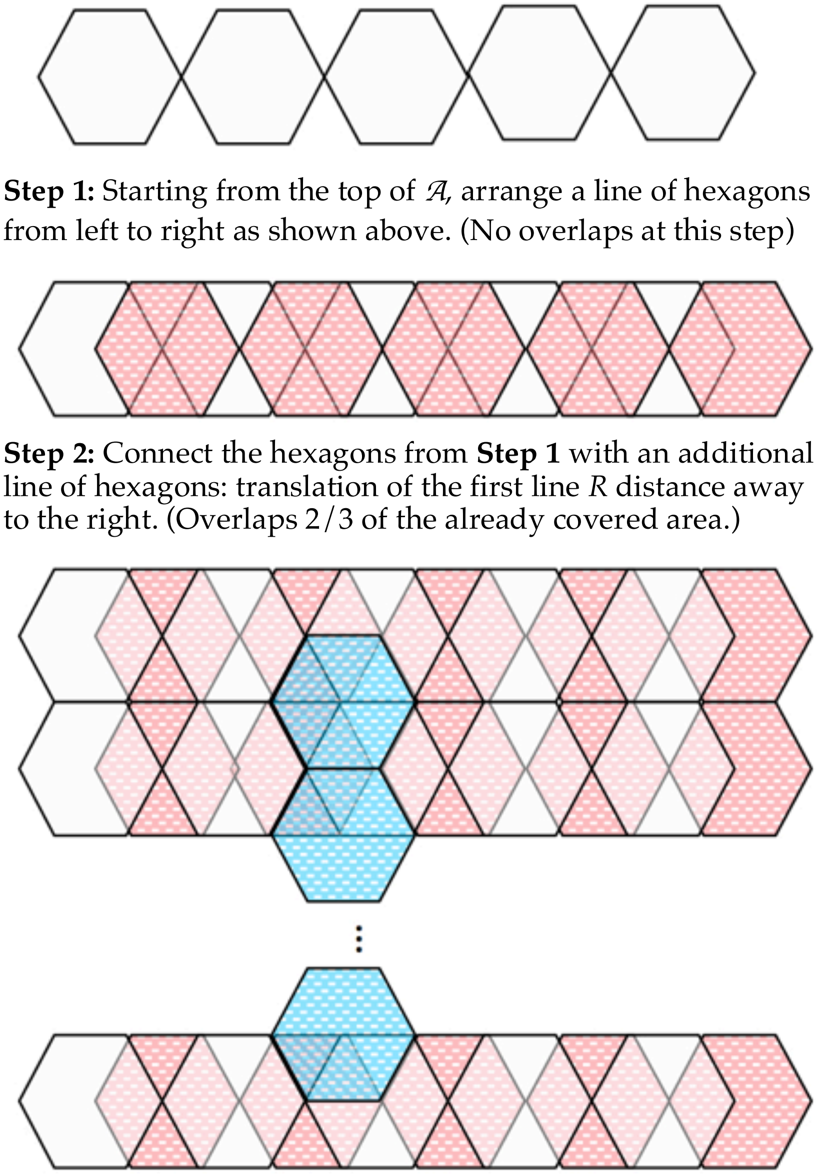
LHP. The LHP results from arranging hexagons in multiple lines. In each line, every two adjacent hexagons share only one common point. A node needs to be picked at every such point so that all nodes covered in each line of hexagons can communicate, partially satisfying iii) of $\Phi(\Lambda(S))$, with least possible resulting overlap between hexagons, satisfying ii) of $\Phi(\Lambda(S))$. It is easy to check that now only $\frac{2}{3}$ of each hexagon's area is covered twice. Yet, at this point nodes covered in line x can communicate only with other nodes covered in x. Placing an additional hexagon for every two adjacent lines is sufficient so that all N nodes communicate (and ii), iii) of $\Phi(\Lambda(S))$ are satisfied as well). Notice that i) of $\Phi(\Lambda(S))$ is still satisfied by the LHP construction at this point. Suppose the area $\mathcal{A}$ is a square of side d and let $q = \frac{\sqrt{|\mathcal{A}|}}{R}$ (assuming without loss of generality $\frac{d}{R}$ is an integer). Asymptotically then, employing LHP, accounting for the $\frac{2}{3}$ overlap and the additional connecting hexagons, one obtains

$$\beta_{min} > k = |X_H| = \left(q^2 + q - \sqrt{3}\right) / \sqrt{3} \tag{D.1}$$

D.2 Upper Bound on the Number of Transmissions

Definition D.1. A *reasonable broadcast* algorithm generates at most $K(q)$ transmissions.

Notice that $K(q)$ is independent on the network density; that is, a reasonable algorithm scales as the number of nodes in $\mathcal{A}$ increases. For instance, flooding is not a reasonable broadcast algorithm by Definition (D.1) since the number of transmissions is proportional to the number of nodes in the network. Notice that a trivial upper bound on the number of retransmissions of all broadcast algorithms (including flooding) could be N: the number of nodes in the network.



Step 1: Starting from the top of $\mathcal{A}$, arrange a line of hexagons from left to right as shown above. (No overlaps at this step)

Step 2: Connect the hexagons from **Step 1** with an additional line of hexagons: translation of the first line R distance away to the right. (Overlaps 2/3 of the already covered area.)

Step 3: Repeat each line of hexagons immediately underneath the previous line, until the area $\mathcal{A}$ is covered completely (aside from edge effects).
&
Step 4: There is no connection between separate lines. After **Step 3**. Therefore, connect the lines with a column of hexagons (as shown in blue).

Figure D.1: Construction of the Linear Hexagon Packing

What is the value of $K(q)$? To answer this, one can attempt to construct the worst case topology which maximizes the size of the minimum connected dominating set.

Suppose one can position the nodes in the network at will. Then, a topology can be constructed that maximizes the minimum needed number of nodes, N, whose transmission disks would cover the entire network area and result in a connected graph. That is, given rectangular area $\mathcal{A}$ and a number of nodes N (not necessarily infinite), one seeks graph with topology, $\Lambda(S)$ such that

$$\beta_{min} = max|X| \tag{D.2}$$

where as before X is the set of nodes picked for transmission while $\Phi(\Lambda(S))$ is satisfied; $\diamond_i = D_i$ here. Notice that the following properties hold given β_{min} and X satisfy (D.2). First, $|X| = N - 1$ (the last covered, N-th, node is not picked to transmit); second, it is guaranteed that the addition of new nodes in the network area would not require additional transmissions, since every new node would be positioned in area already covered by one of the $N - 1$ nodes in X; and third, if N is only dependent on $\mathcal{A}$'s dimensions and R, then $N - 1$ is equal to $K(q)$ by Definition (D.1).

In the previous section, linear arrangement of hexagons was employed to construct an approximation of the lower bound on the number of transmissions. Here, to construct $\Lambda(S)$ a different hexagons' property is utilized.

In 1940, L. Fejes Tóth [191] proved that the densest packing of circles (or any collection of other shapes given each shape encloses the same surface area) in the plane is obtained by the honeycomb hexagonal lattice.

Suppose the plane is tessellated by hexagons. Then, the honeycomb circle

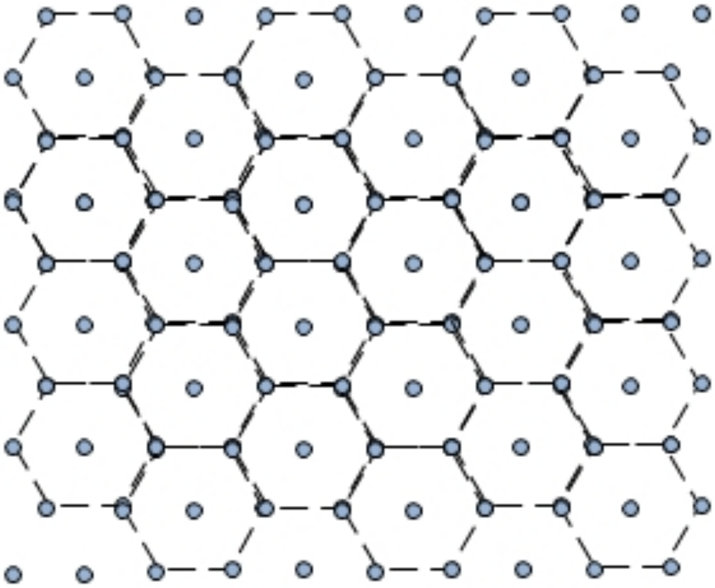


Figure D.2: Honeycomb-based topology: nodes are placed at the vertices and centers of tessellating hexagons of side $r\sqrt{3}$.

packing requires that each packed circle is centered at the centroid of a tessellating hexagon. The largest number of packed circles results from Tóth's result. Figure D.2 demonstrates the union of 7 such circle packings (the center of each circle in the union of packings is represented by a node in fig. D.2). Only one of the underlying 7 honeycomb lattices (dashed line) is shown in fig. D.2 for clarity. Again, here is is assumed, without loss of generality, that the rectangular area is a square with side d and $|\mathcal{A}| = d^2$; the radius of a circle is equal to the transmission radius R, but the side of a tessellating hexagon this time is equal to $3\sqrt{R}$.

Suppose each node in fig. D.2 is replaced by a pair of nodes separated by distance ϵ, where $\epsilon \to 0$.

Consider the transmissions' row in fig. D.3. The removal of any node from the row would disconnect the graph. Note that at this point, nodes in row x of

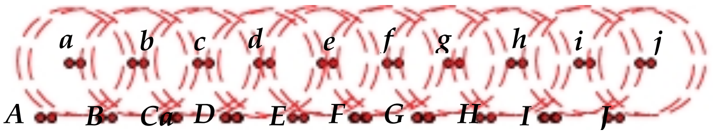


Figure D.3: Removal of any node from pairs a through j would render the communication between nodes in the broadcast row impossible. Note that the nodes from pairs A through J are not covered by the transmissions of nodes in pairs a though j.

the hexagonal topology are not connected to nodes in rows other than x (i.e., the rows form connected components in the graph). To connect the rows into single connected component, additional nodes need to be placed. As above, one would like to maximize the minimum number of connecting nodes needed.

Figure D.4 demonstrates the final graph topology $\Lambda(S)$ including the additional connecting pairs of nodes (in green). Note that eliminating any node's transmission (except of the last covered node) would cause disconnections in the graph. Hence, $N-1$ nodes need to transmit so that condition iii) of $\Phi(\Lambda(S))$ is satisfied; thus indeed $|X| = N-1$. Given $N-1$ transmitting nodes i) and ii) of $\Phi(\Lambda(S))$ are also satisfied; ii) is satisfied since due to symmetry, one cannot select a different subset X of the N nodes, such that $\bigcap_{i \in X} \diamond_i$ is less.

By construction, $\Lambda(S)$ maximizes the minimum number of nodes needed whose transmission disks would cover the entire network area and result in connected graph. Initially there are $R-1$ pairs per row; adding the nodes needed to connect every two rows yields $R+1$ pairs of nodes per row. Since the rows are distance R from one another, there are $R-1$ rows. Thus, at most

$$\beta_{min} = |X| = N-1 = 2(q+1)(q-1) - 1 = 2(q^2-1) - 1 \qquad \text{(D.3)}$$

transmissions are needed for network coverage. Hence, $|X|$ is independent of the network density. By Definition (D.1), $K(q) = |X| = \beta_{min}$, providing an upper

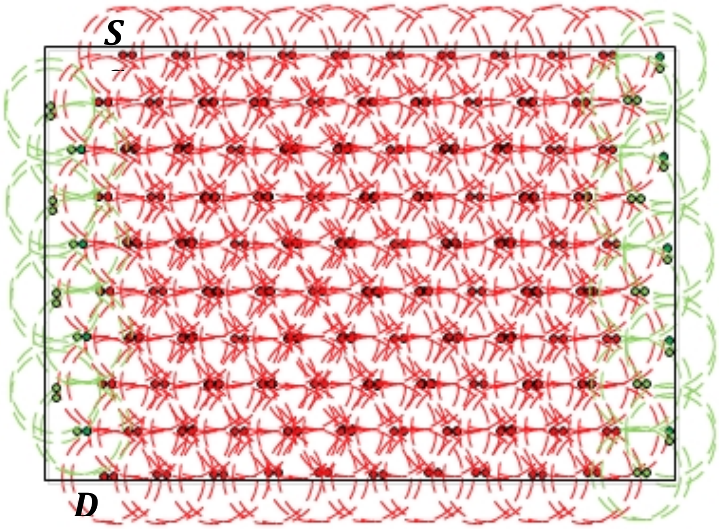


Figure D.4: Nodes in green connect broadcast rows so that all N nodes communicate. Broadcast starting from node $\boldsymbol{S}$ proceeds in a "zig-zag" fashion until node $\boldsymbol{D}$ is reached. All N nodes are required to transmit: $|MCS\,D|$=N. Removal of any node would cause disconnection. The network area (black rectangle) is completely covered by the transmissions.

bound on the number of needed transmissions by any *reasonable* algorithm to cover the network. K is also the lowest number of transmissions any algorithm can guarantee given a network area with dimensions $d \times d$ and transmission radius of R. In contrast, the lowest number of transmissions any algorithm can ever achieve as the density of the network grows to infinity is approximated by (D.1).

BIBLIOGRAPHY

[1] "The Zettabyte Era - Trends and Analysis." http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/VNI_Hyperconnectivity_WP.html#_ftnref3.

[2] S. Lloyd, "Ultimate physical limits to computation," *Nature*, vol. 406, pp. 1047–1054, 2000.

[3] M. Hilbert and P. Lpez, "The worlds technological capacity to store, communicate, and compute information," *Science*, vol. 332, no. 6025, pp. 60–65, 2011.

[4] "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013 - 2018." http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html.

[5] 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," *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, vol. 3, pp. 522–535, Dec 2010.

[6] S. Dawson-Haggerty, S. Lanzisera, J. Taneja, R. Brown, and D. Culler, "@scale: Insights from a large, long-lived appliance energy wsn," in *Proceedings of the 11th International Conference on Information Processing in Sensor Networks*, IPSN '12, (New York, NY, USA), pp. 37–48, ACM, 2012.

[7] I. Dietrich and F. Dressler, "On the lifetime of wireless sensor networks," *ACM Trans. Sen. Netw.*, vol. 5, pp. 5:1–5:39, Feb. 2009.

[8] V. Shnayder, M. Hempstead, B.-r. Chen, G. W. Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*, SenSys '04, (New York, NY, USA), pp. 188–200, ACM, 2004.

[9] O. Landsiedel, K. Wehrle, and S. Gotz, "Accurate prediction of power consumption in sensor networks," in *Embedded Networked Sensors, 2005. EmNetS-II. The Second IEEE Workshop on*, pp. 37–44, May 2005.

[10] T. Pering, Y. Agarwal, R. Gupta, and R. Want, "Coolspots: Reducing the power consumption of wireless mobile devices with multiple radio interfaces," in *Proceedings of the 4th International Conference on Mobile Systems, Applications and Services*, MobiSys '06, (New York, NY, USA), pp. 220–232, ACM, 2006.

[11] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, MobiCom '99, (New York, NY, USA), pp. 151–162, ACM, 1999.

[12] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theor.*, vol. 19, pp. 471–480, Sept. 2006.

[13] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, "Networked slepian-wolf: theory, algorithms, and scaling laws," *Information Theory, IEEE Transactions on*, vol. 51, pp. 4057–4073, Dec 2005.

[14] K. Yuen, B. Liang, and B. Li, "A distributed framework for correlated data gathering in sensor networks," *Vehicular Technology, IEEE Transactions on*, vol. 57, pp. 578–593, Jan 2008.

[15] S. Pradhan, J. Kusuma, and K. Ramchandran, "Distributed compression in a dense microsensor network," *Signal Processing Magazine, IEEE*, vol. 19, pp. 51–60, Mar 2002.

[16] T. M. Cover and J. A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006.

[17] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *Comm. Mag.*, vol. 40, pp. 102–114, Aug. 2002.

[18] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Commun. ACM*, vol. 43, pp. 51–58, May 2000.

[19] J. M. Rabaey, M. J. Ammer, J. L. da Silva, D. Patel, and S. Roundy, "Picoradio supports ad hoc ultra-low power wireless networking," *Computer*, vol. 33, pp. 42–48, July 2000.

[20] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the*

1st ACM International Workshop on Wireless Sensor Networks and Applications, WSNA '02, (New York, NY, USA), pp. 88–97, ACM, 2002.

[21] M. C. Vuran and I. Akyildiz, "Spatial correlation-based collaborative medium access control in wireless sensor networks," *Networking, IEEE/ACM Transactions on*, vol. 14, pp. 316–329, April 2006.

[22] M. Gastpar and M. Vetterli, "Power, spatio-temporal bandwidth, and distortion in large sensor networks," *Selected Areas in Communications, IEEE Journal on*, vol. 23, pp. 745–754, April 2005.

[23] S. Bandyopadhyay, Q. Tian, and E. Coyle, "Spatio-temporal sampling rates and energy efficiency in wireless sensor networks," *Networking, IEEE/ACM Transactions on*, vol. 13, pp. 1339–1352, Dec 2005.

[24] M. Cardei, M. Thai, Y. Li, and W. Wu, "Energy-efficient target coverage in wireless sensor networks," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 3, pp. 1976–1984 vol. 3, March 2005.

[25] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *Information Theory, IEEE Transactions on*, vol. 52, pp. 489–509, Feb 2006.

[26] D. Donoho, "Compressed sensing," *Information Theory, IEEE Transactions on*, vol. 52, pp. 1289–1306, April 2006.

[27] E. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?," *Information Theory, IEEE Transactions on*, vol. 52, pp. 5406–5425, Dec 2006.

[28] E. Candes and M. Wakin, "An introduction to compressive sampling," *Signal Processing Magazine, IEEE*, vol. 25, pp. 21–30, March 2008.

[29] X. Wu, Q. Wang, and M. Liu, "In-situ soil moisture sensing: Measurement scheduling and estimation using sparse sampling," *ACM Trans. Sen. Netw.*, vol. 11, pp. 26:1–26:29, July 2014.

[30] S. Lee, S. Pattem, M. Sathiamoorthy, B. Krishnamachari, and A. Ortega, "Spatially-localized compressed sensing and routing in multi-hop sensor networks," in *Proceedings of the 3rd International Conference on GeoSensor Networks*, GSN '09, (Berlin, Heidelberg), pp. 11–20, Springer-Verlag, 2009.

[31] J. Wang, S. Tang, B. Yin, and X.-Y. Li, "Data gathering in wireless sensor networks through intelligent compressive sensing," in *INFOCOM, 2012 Proceedings IEEE*, pp. 603–611, March 2012.

[32] C. Luo, F. Wu, J. Sun, and C. W. Chen, "Compressive data gathering for large-scale wireless sensor networks," in *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking*, MobiCom '09, (New York, NY, USA), pp. 145–156, ACM, 2009.

[33] D. Baron, M. F. Duarte, M. B. Wakin, S. Sarvotham, and R. G. Baraniuk, "Distributed compressive sensing," *CoRR*, vol. abs/0901.3403, 2009.

[34] R. Mudumbai, J. Hespanha, U. Madhow, and G. Barriac, "Distributed transmit beamforming using feedback control," *Information Theory, IEEE Transactions on*, vol. 56, pp. 411–426, Jan 2010.

[35] A. Bletsas, A. Lippman, and J. Sahalos, "Simple, zero-feedback, distributed beamforming with unsynchronized carriers," *Selected Areas in Communications, IEEE Journal on*, vol. 28, pp. 1046–1054, September 2010.

[36] W. Bajwa, J. Haupt, A. Sayeed, and R. Nowak, "Compressive wireless sensing," in *Information Processing in Sensor Networks, 2006. IPSN 2006. The Fifth International Conference on*, pp. 134–142, 2006.

[37] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, vol. 4, pp. 2033–2036 vol.4, 2001.

[38] L. Li, J. Y. Halpern, P. Bahl, Y.-M. Wang, and R. Wattenhofer, "Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks," in *Proceedings of the Twentieth Annual ACM Symposium on Principles of Distributed Computing*, PODC '01, (New York, NY, USA), pp. 264–273, ACM, 2001.

[39] X. Cheng, B. Narahari, R. Simha, M. Cheng, and D. Liu, "Strong minimum energy topology in wireless sensor networks: Np-completeness and heuristics," *Mobile Computing, IEEE Transactions on*, vol. 2, pp. 248–256, July 2003.

[40] E. L. Lloyd, R. Liu, M. V. Marathe, R. Ramanathan, and S. S. Ravi, "Algorithmic aspects of topology control problems for ad hoc networks," *Mob. Netw. Appl.*, vol. 10, pp. 19–34, Feb. 2005.

[41] R. Ramanathan and R. Rosales-Hain, “Topology control of multihop wireless networks using transmit power adjustment,” in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, pp. 404–413 vol.2, 2000.

[42] J. Pan, Y. T. Hou, L. Cai, Y. Shi, and S. X. Shen, “Topology control for wireless sensor networks,” in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, MobiCom ’03, (New York, NY, USA), pp. 286–299, ACM, 2003.

[43] S. Misra, S. D. Hong, G. Xue, and J. Tang, “Constrained relay node placement in wireless sensor networks: Formulation and approximations,” *Networking, IEEE/ACM Transactions on*, vol. 18, pp. 434–447, April 2010.

[44] Y. Hou, Y. Shi, H. Sherali, and S. Midkiff, “On energy provisioning and relay node placement for wireless sensor networks,” *Wireless Communications, IEEE Transactions on*, vol. 4, pp. 2579–2590, Sept 2005.

[45] J. Bredin, E. Demaine, M. Hajiaghayi, and D. Rus, “Deploying sensor networks with guaranteed fault tolerance,” *Networking, IEEE/ACM Transactions on*, vol. 18, pp. 216–228, Feb 2010.

[46] A. Kashyap, S. Khuller, and M. Shayman, “Relay placement for higher order connectivity in wireless sensor networks,” in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pp. 1–12, April 2006.

[47] H. Liu, P. jun Wan, and X. Jia, “Fault-tolerant relay node placement in wireless sensor networks,” *LNCS*, vol. 3595, pp. 230–239, 2005.

[48] S. Misra, S. D. Hong, G. Xue, and J. Tang, “Constrained relay node placement in wireless sensor networks to meet connectivity and survivability requirements,” in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pp. –, April 2008.

[49] J. Tang, B. Hao, and A. Sen, “Relay node placement in large scale wireless sensor networks,” *Comput. Commun.*, vol. 29, pp. 490–501, Feb. 2006.

[50] G.-H. Lin and G. Xue, “Steiner tree problem with minimum number of steiner points and bounded edge-length,” *Inf. Process. Lett.*, vol. 69, pp. 53–57, Jan. 1999.

[51] R. Ravi and D. Williamson, "An approximation algorithm for minimum-cost vertex-connectivity problems," *Algorithmica*, vol. 18, no. 1, pp. 21–43, 1997.

[52] I. Rubin, A. Behzad, R. Zhang, H. Luo, and E. Caballero, "Tbone: A mobile-backbone protocol for ad hoc wireless networks," in *Aerospace Conference Proceedings, 2002. IEEE*, vol. 6, pp. 6–2727–6–2740 vol.6, 2002.

[53] K. Xu, X. Hong, and M. Gerla, "Landmark routing in ad hoc networks with mobile backbones," *J. Parallel Distrib. Comput.*, vol. 63, pp. 110–122, Feb. 2003.

[54] A. Srinivas, G. Zussman, and E. Modiano, "Mobile backbone networks –: Construction and maintenance," in *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '06, (New York, NY, USA), pp. 166–177, ACM, 2006.

[55] F. El-Moukaddem, E. Torng, G. Xing, E. Torng, G. Xing, and G. Xing, "Mobile relay configuration in data-intensive wireless sensor networks," *Mobile Computing, IEEE Transactions on*, vol. 12, pp. 261–273, Feb 2013.

[56] W. Wang, V. Srinivasan, and K.-C. Chua, "Using mobile relays to prolong the lifetime of wireless sensor networks," in *Proceedings of the 11th Annual International Conference on Mobile Computing and Networking*, MobiCom '05, (New York, NY, USA), pp. 270–283, ACM, 2005.

[57] D. K. Goldenberg, J. Lin, A. S. Morse, B. E. Rosen, and Y. R. Yang, "Towards mobility as a network control primitive," in *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '04, (New York, NY, USA), pp. 163–174, ACM, 2004.

[58] A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg, "Near-optimal sensor placements: Maximizing information while minimizing communication cost," in *Proceedings of the 5th International Conference on Information Processing in Sensor Networks*, IPSN '06, (New York, NY, USA), pp. 2–10, ACM, 2006.

[59] Y. Chen and A. Terzis, "On the implications of the log-normal path loss model: An efficient method to deploy and move sensor motes," in *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, SenSys '11, (New York, NY, USA), pp. 26–39, ACM, 2011.

[60] E. Flushing and G. Di Caro, “Relay node placement for performance enhancement with uncertain demand: A robust optimization approach,” in *Modeling Optimization in Mobile, Ad Hoc Wireless Networks (WiOpt), 2013 11th International Symposium on*, pp. 556–563, May 2013.

[61] V. Shah and S. Krishnamurthy, “Handling asymmetry in power heterogeneous ad hoc networks: A cross layer approach,” in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*, ICDCS ’05, (Washington, DC, USA), pp. 749–759, IEEE Computer Society, 2005.

[62] L. Sang, A. Arora, and H. Zhang, “On link asymmetry and one-way estimation in wireless sensor networks,” *ACM Trans. Sen. Netw.*, vol. 6, pp. 12:1–12:25, Mar. 2010.

[63] E. Hua and Z. Haas, “An algorithm for prediction of link lifetime in manet based on unscented kalman filter,” *Communications Letters, IEEE*, vol. 13, pp. 782–784, October 2009.

[64] M. Hu, Z. Zhong, H. Zhu, M. Ni, and C.-Y. Chang, “Analytical modeling of link duration for vehicular ad hoc networks in urban environment,” in *Proceeding of the Tenth ACM International Workshop on Vehicular Internetworking, Systems, and Applications*, VANET ’13, (New York, NY, USA), pp. 61–70, ACM, 2013.

[65] J. T. Moy, *OSPF: Anatomy of an Internet Routing Protocol*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1998.

[66] S. Kumar, V. S. Raghavan, and J. Deng, “Medium access control protocols for ad hoc wireless networks: A survey,” *Ad Hoc Netw.*, vol. 4, pp. 326–358, May 2006.

[67] L. Shengbin and Z. Xiaoliang, “A survey on mac layer in ad hoc wireless networks,” in *Applied Informatics and Communication* (D. Zeng, ed.), vol. 225 of *Communications in Computer and Information Science*, pp. 691–699, Springer Berlin Heidelberg, 2011.

[68] S. Jain, K. Fall, and R. Patra, “Routing in a delay tolerant network,” in *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM ’04, (New York, NY, USA), pp. 145–158, ACM, 2004.

[69] A. Vahdat and D. Becker, “Epidemic routing for partially-connected ad hoc networks,” tech. rep., Duke University, 2000.

[70] C.-K. Toh, “Associativity-based routing for ad hoc mobile networks,” *Wirel. Pers. Commun.*, vol. 4, pp. 103–139, Mar. 1997.

[71] D. B. Johnson, D. A. Maltz, and J. Broch, “Ad hoc networking,” in *Ad Hoc Networking*, ch. DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks, pp. 139–172, Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2001.

[72] C. Perkins and E. Royer, “Ad-hoc on-demand distance vector routing,” in *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, pp. 90–100, Feb 1999.

[73] V. D. Park and M. S. Corson, “A highly adaptive distributed routing algorithm for mobile wireless networks,” in *Proceedings of the INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution*, INFOCOM '97, (Washington, DC, USA), pp. 1405–, IEEE Computer Society, 1997.

[74] C. E. Perkins and P. Bhagwat, “Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers,” in *Proceedings of the Conference on Communications Architectures, Protocols and Applications*, SIGCOMM '94, (New York, NY, USA), pp. 234–244, ACM, 1994.

[75] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, “Optimized link state routing protocol for ad hoc networks,” in *Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International*, pp. 62–68, 2001.

[76] P. Krishna, N. H. Vaidya, M. Chatterjee, and D. K. Pradhan, “A cluster-based approach for routing in dynamic networks,” *SIGCOMM Comput. Commun. Rev.*, vol. 27, pp. 49–64, Apr. 1997.

[77] Z. Haas, “A new routing protocol for the reconfigurable wireless networks,” in *Universal Personal Communications Record, 1997. Conference Record., 1997 IEEE 6th International Conference on*, vol. 2, pp. 562–566 vol.2, Oct 1997.

[78] H. Hassanein and A. Zhou, “Routing with load balancing in wireless ad hoc networks,” in *Proceedings of the 4th ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, MSWIM '01, (New York, NY, USA), pp. 89–96, ACM, 2001.

[79] L. Popa, A. Rostamizadeh, R. Karp, C. Papadimitriou, and I. Stoica, "Balancing traffic load in wireless networks with curveball routing," in *Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '07, (New York, NY, USA), pp. 170–179, ACM, 2007.

[80] K. W. Choi, W. S. Jeon, and D. G. Jeong, "Efficient load-aware routing scheme for wireless mesh networks," *Mobile Computing, IEEE Transactions on*, vol. 9, pp. 1293–1307, Sept 2010.

[81] M. Gerla, C.-C. Chiang, and L. Zhang, "Tree multicast strategies in mobile, multishop wireless networks," *Mob. Netw. Appl.*, vol. 4, pp. 193–207, Oct. 1999.

[82] S. E. Deering and D. R. Cheriton, "Multicast routing in datagram internetworks and extended lans," *ACM Trans. Comput. Syst.*, vol. 8, pp. 85–110, May 1990.

[83] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1979.

[84] F. Bouabdallah, N. Bouabdallah, and R. Boutaba, "Toward reliable and efficient reporting in wireless sensor networks," *Mobile Computing, IEEE Transactions on*, vol. 7, pp. 978–994, Aug 2008.

[85] A. Khandani, J. Abounadi, E. Modiano, and L. Zheng, "Cooperative routing in static wireless networks," *Communications, IEEE Transactions on*, vol. 55, pp. 2185–2192, Nov 2007.

[86] M. Elhawary and Z. Haas, "Energy-efficient protocol for cooperative networks," *Networking, IEEE/ACM Transactions on*, vol. 19, pp. 561–574, April 2011.

[87] M. Nikolov and Z. J. Haas, "Encoded sensing for energy efficient wireless networks," in *ACM Transactions on Sensor Networks*, ACM, in submission.

[88] M. Nikolov and Z. J. Haas, "Collaborating with correlation for energy efficient wsn," in *Proceedings of the First ACM International Workshop on Mission-oriented Wireless Sensor Networking*, MiSeNet '12, (New York, NY, USA), pp. 27–32, ACM, 2012.

[89] K. R. Varshney and P. M. van de Ven, "Balancing lifetime and classification accuracy of wireless sensor networks," in *Proceedings of the Fourteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '13, (New York, NY, USA), pp. 31–38, ACM, 2013.

[90] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. New York, NY, USA: Cambridge University Press, 2005.

[91] C. Chien, I. Elgorriaga, and C. McConaghy, "Low-power direct-sequence spread-spectrum modem architecture for distributed wireless sensor networks," in *Proceedings of the 2001 International Symposium on Low Power Electronics and Design*, ISLPED '01, (New York, NY, USA), pp. 251–254, ACM, 2001.

[92] S. Lingyang and J. Shen, *Evolved cellular network planning and optimization for UMTS and LTE*. Hoboken, NJ: CRC Press, 2010.

[93] A. J. Viterbi, *CDMA: Principles of Spread Spectrum Communication*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 1995.

[94] U. Madhow and M. B. Pursley, "Acquisition in direct-sequence spread-spectrum communication networks: An asymptotic analysis," *IEEE Trans. Inf. Theor.*, vol. 39, pp. 903–912, May 1993.

[95] D. Lehmer, "The machine tools of combinatorics.," in *The machine tools of combinatorics.* (E. Beckenback, ed.), Applied Combinatorial Mathematics, pp. 5–31, Wiley, NY, 1964.

[96] G. L. Stüber, *Principles of Mobile Communication (2Nd Ed.)*. Norwell, MA, USA: Kluwer Academic Publishers, 2001.

[97] A. Goldsmith, *Wireless Communications*. New York, NY, USA: Cambridge University Press, 2005.

[98] R. Baraniuk, "Compressive sensing [lecture notes]," *Signal Processing Magazine, IEEE*, vol. 24, pp. 118–121, July 2007.

[99] J. Tropp and A. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *Information Theory, IEEE Transactions on*, vol. 53, pp. 4655–4666, Dec 2007.

[100] S. Verdu, “Minimum probability of error for asynchronous gaussian multiple-access channels,” *IEEE Trans. Inf. Theor.*, vol. 32, pp. 85–96, Jan. 1986.

[101] H. V. Poor and S. Verdu, “Probability of error in mmse multiuser detection,” *IEEE Trans. Inf. Theor.*, vol. 43, pp. 858–871, Sept. 2006.

[102] P. D. Robinson and A. J. Wathen, “Variational bounds on the entries of the inverse of a matrix,” *IMA Journal of Numerical Analysis*, vol. 12, no. 4, pp. 463–486, 1992.

[103] G. H. Golub and C. F. Van Loan, *Matrix Computations (3rd Ed.)*. Baltimore, MD, USA: Johns Hopkins University Press, 1996.

[104] M. Chiani, D. Dardari, and M. K. Simon, “New exponential bounds and approximations for the computation of error probability in fading channels,” *Wireless Communications, IEEE Transactions on*, vol. 2, pp. 840–845, July 2003.

[105] J. O. Berger, V. D. Oliveira, and B. Sanso, “Objective bayesian analysis of spatially correlated data,” *JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION*, vol. 96, pp. 1361–1374, 2000.

[106] Paul, *Bayesian Logic (BLOG) Inference Engine.* Retrieved June 1, 2014 from `http://people.csail.mit.edu/milch/blog/index.html`.

[107] R. Rick and L. Milstein, “Noncoherent parallel acquisition in cdma spread spectrum systems,” in *Communications, 1994. ICC '94, SUPERCOMM/ICC '94, Conference Record, 'Serving Humanity Through Communications.' IEEE International Conference on*, pp. 1422–1426 vol.3, May 1994.

[108] A. Nosratinia, T. Hunter, and A. Hedayat, “Cooperative communication in wireless networks,” *Communications Magazine, IEEE*, vol. 42, pp. 74–80, Oct 2004.

[109] S. Cui, A. Goldsmith, and A. Bahai, “Energy-efficiency of mimo and cooperative mimo techniques in sensor networks,” *Selected Areas in Communications, IEEE Journal on*, vol. 22, pp. 1089–1098, Aug 2004.

[110] A. Sendonaris, E. Erkip, and B. Aazhang, “User cooperation diversity. part i. system description,” *Communications, IEEE Transactions on*, vol. 51, pp. 1927–1938, Nov 2003.

[111] J. Laneman, D. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *Information Theory, IEEE Transactions on*, vol. 50, pp. 3062–3080, Dec 2004.

[112] L. Simic, S. Berber, and K. Sowerby, "Partner choice and power allocation for energy efficient cooperation in wireless sensor networks," in *Communications, 2008. ICC '08. IEEE International Conference on*, pp. 4255–4260, May 2008.

[113] X. Li, M. Chen, and W. Liu, "Application of stbc-encoded cooperative transmissions in wireless sensor networks," *Signal Processing Letters, IEEE*, vol. 12, pp. 134–137, Feb 2005.

[114] D. Wu, Y. Cai, L. Zhou, and J. Wang, "A cooperative communication scheme based on coalition formation game in clustered wireless sensor networks," *Wireless Communications, IEEE Transactions on*, vol. 11, pp. 1190–1200, March 2012.

[115] R. Mudumbai, D. Brown, U. Madhow, and H. Poor, "Distributed transmit beamforming: challenges and recent progress," *Communications Magazine, IEEE*, vol. 47, pp. 102–110, February 2009.

[116] M. A. Razzaque, C. Bleakley, and S. Dobson, "Compression in wireless sensor networks: A survey and comparative evaluation," *ACM Trans. Sen. Netw.*, vol. 10, pp. 5:1–5:44, Dec. 2013.

[117] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theor.*, vol. 46, pp. 388–404, Sept. 2006.

[118] Y. Zhang and M. Quilling, "Optimal backbone generation for robotic relay networks," in *Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on*, pp. 1–6, July 2011.

[119] Z. Haas and R. Barr, "Density-independent, scalable search in ad hoc networks," in *Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005. IEEE 16th International Symposium on*, vol. 2, pp. 1401–1408 Vol. 2, Sept 2005.

[120] T. Rappaport, *Wireless Communications: Principles and Practice*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2nd ed., 2001.

[121] M. Zuniga and B. Krishnamachari, "Analyzing the transitional region in

low power wireless links," in *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pp. 517–526, Oct 2004.

[122] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[123] D. Zuckerman, "Linear degree extractors and the inapproximability of max clique and chromatic number," in *Proceedings of the Thirty-eighth Annual ACM Symposium on Theory of Computing*, STOC '06, (New York, NY, USA), pp. 681–690, ACM, 2006.

[124] K. Kashiwabara, Y. Okamoto, and T. Uno, "Matroid representation of clique complexes," in *Proceedings of the 9th Annual International Conference on Computing and Combinatorics*, COCOON'03, (Berlin, Heidelberg), pp. 192–201, Springer-Verlag, 2003.

[125] M. Fisher, G. Nemhauser, and L. Wolsey, "An analysis of approximations for maximizing submodular set functionsii," in *Polyhedral Combinatorics* (M. Balinski and A. Hoffman, eds.), vol. 8 of *Mathematical Programming Studies*, pp. 73–87, Springer Berlin Heidelberg, 1978.

[126] G. Greiner and K. Hormann, "Efficient clipping of arbitrary polygons," *ACM Trans. Graph.*, vol. 17, pp. 71–83, Apr. 1998.

[127] A. Gupta, A. Roth, G. Schoenebeck, and K. Talwar, "Constrained non-monotone submodular maximization: Offline and secretary algorithms," in *Proceedings of the 6th International Conference on Internet and Network Economics*, WINE'10, (Berlin, Heidelberg), pp. 246–257, Springer-Verlag, 2010.

[128] J. Lee, V. S. Mirrokni, V. Nagarajan, and M. Sviridenko, "Non-monotone submodular maximization under matroid and knapsack constraints," in *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*, STOC '09, (New York, NY, USA), pp. 323–332, ACM, 2009.

[129] X. Cheng, D.-Z. Du, L. Wang, and B. Xu, "Relay sensor placement in wireless sensor networks," *Wirel. Netw.*, vol. 14, pp. 347–355, June 2008.

[130] E. Lloyd and G. Xue, "Relay node placement in wireless sensor networks," *Computers, IEEE Transactions on*, vol. 56, pp. 134–138, Jan 2007.

[131] S. Li, G. Chen, and W. Ding, "Relay node placement in heterogeneous

wireless sensor networks with basestations," in *Communications and Mobile Computing, 2009. CMC '09. WRI International Conference on*, vol. 1, pp. 573–577, Jan 2009.

[132] W. Zhang, G. Xue, and S. Misra, "Fault-tolerant relay node placement in wireless sensor networks: Problems and algorithms," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pp. 1649–1657, May 2007.

[133] M. Patel, R. Chandrasekaran, and S. Venkatesan, "Energy efficient sensor, relay and base station placements for coverage, connectivity and routing," in *IPCCC*, pp. 581–586, IEEE, 06 2007.

[134] A. Kashyap, F. Sun, and M. Shayman, "Relay placement for minimizing congestion in wireless backbone networks," in *Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE*, vol. 1, pp. 159–164, April 2006.

[135] S. Milner, J. Llorca, and C. Davis, "Autonomous reconfiguration and control in directional mobile ad hoc networks," *Circuits and Systems Magazine, IEEE*, vol. 9, pp. 10–26, Second 2009.

[136] M. Khabbazian, I. F. Blake, and V. K. Bhargava, "Local broadcast algorithms in wireless ad hoc networks: Reducing the number of transmissions," *Mobile Computing, IEEE Transactions on*, vol. 11, no. 3, pp. 402–413, 2012.

[137] J. Lu and K. Whitehouse, "Flash flooding: Exploiting the capture effect for rapid flooding in wireless sensor networks," in *INFOCOM 2009, IEEE*, pp. 2491–2499, IEEE, 2009.

[138] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh, "Efficient network flooding and time synchronization with glossy," in *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on*, pp. 73–84, April 2011.

[139] B. N. Clark, C. J. Colbourn, and D. S. Johnson, "Unit disk graphs," *Annals of Discrete Mathematics*, vol. 48, pp. 165–177, 1991.

[140] Z. J. Haas, J. Y. Halpern, and L. Li, "Gossip-based ad hoc routing," *IEEE/ACM Trans. Netw.*, vol. 14, pp. 479–491, June 2006.

[141] Y. Sasson, D. Cavin, and A. Schiper, “Probabilistic broadcast for flooding in wireless mobile ad hoc networks,” in *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, vol. 2, pp. 1124–1130 vol.2, March 2003.

[142] D. J. Scott and A. Yasinsac, “Dynamic probabilistic retransmission in ad hoc networks,” in *Proceedings of the International Conference on Wireless Networks, ICWN '04, June 21-24, 2004, Las Vegas, Nevada, USA, Volume 1*, pp. 158–164, 2004.

[143] J. Cartigny, D. Simplot, and J. Carle, “Stochastic flooding broadcast protocols in mobile wireless networks,” *LIFL Univ. Lille*, vol. 1, 2002.

[144] J. soo Kim, Q. Zhang, and D. Agrawal, “Probabilistic broadcasting based on coverage area and neighbor confirmation in mobile ad hoc networks,” in *Global Telecommunications Conference Workshops, 2004. GlobeCom Workshops 2004. IEEE*, pp. 96–101, Nov 2004.

[145] A. Keshavarz-Haddad, V. Ribeiro, and R. Riedi, “Color-based broadcasting for ad hoc networks,” in *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2006 4th International Symposium on*, pp. 1–10, IEEE, 2006.

[146] S. Pleisch, M. Balakrishnan, K. Birman, and R. van Renesse, “Mistral: Efficient flooding in mobile ad-hoc networks,” in *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '06, (New York, NY, USA), pp. 1–12, ACM, 2006.

[147] F. Stann, J. Heidemann, R. Shroff, and M. Z. Murtaza, “Rbp: Robust broadcast propagation in wireless networks,” in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, SenSys '06, (New York, NY, USA), pp. 85–98, ACM, 2006.

[148] S. Basagni, M. Mastrogiovanni, A. Panconesi, and C. Petrioli, “Localized protocols for ad hoc clustering and backbone formation: A performance comparison,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 17, pp. 292–306, Apr. 2006.

[149] P.-J. Wan, K. Alzoubi, and O. Frieder, “Distributed construction of connected dominating set in wireless ad hoc networks,” in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, pp. 1597–1604 vol.3, 2002.

[150] S. Funke, A. Kesselman, U. Meyer, and M. Segal, “A simple improved distributed algorithm for minimum cds in unit disk graphs,” in *Wireless And Mobile Computing, Networking And Communications, 2005. (WiMob'2005), IEEE International Conference on*, vol. 2, pp. 220–223 Vol. 2, Aug 2005.

[151] B. Gao, Y. Yang, and H. Ma, “An effective distributed approximation algorithm for constructing minimum connected dominating set in wireless ad hoc networks,” in *Computer and Information Technology, 2004. CIT '04. The Fourth International Conference on*, pp. 658–663, Sept 2004.

[152] X. Cheng, X. Huang, D. Li, W. Wu, and D.-Z. Du, “A polynomial-time approximation scheme for the minimum-connected dominating set in ad hoc wireless networks,” *Networks*, vol. 42, no. 4, pp. 202–208, 2003.

[153] H. B. Hunt III, M. V. Marathe, V. Radhakrishnan, S. S. Ravi, D. J. Rosenkrantz, and R. E. Stearns, “Nc-approximation schemes for np- and pspace-hard problems for geometric graphs,” *Journal of Algorithms*, vol. 26, no. 2, pp. 238–274, 1998.

[154] M. Khabbazian and V. K. Bhargava, “Efficient broadcasting in mobile ad hoc networks,” *Mobile Computing, IEEE Transactions on*, vol. 8, no. 2, pp. 231–245, 2009.

[155] M. Nikolov and Z. J. Haas, “Towards optimal broadcast in wireless networks,” in *IEEE Transactions on Mobile Computing*, IEEE, September 2014.

[156] R. Solis, V. Borkar, and P. Kumar, “A new distributed time synchronization protocol for multihop wireless networks,” in *Decision and Control, 2006 45th IEEE Conference on*, pp. 2734–2739, Dec 2006.

[157] R. Dokania, X. Wang, W. Godycki, C. Dorta-Quinones, and A. Apsel, “Pco based event propagation scheme for globally synchronized sensor networks,” in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pp. 1–5, Dec 2010.

[158] S. Guha and S. Khuller, “Approximation algorithms for connected dominating sets,” *Algorithmica*, vol. 20, pp. 374–387, Apr. 1998.

[159] Y. Cai, K. Hua, and A. Phillips, “Leveraging 1-hop neighborhood knowledge for efficient flooding in wireless ad hoc networks,” in *Performance, Computing, and Communications Conference, 2005. IPCCC 2005. 24th IEEE International*, pp. 347–354, April 2005.

[160] H. Liu, X. Jia, P.-J. Wan, X. Liu, and F. F. Yao, "A distributed and efficient flooding scheme using 1-hop information in mobile ad hoc networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, pp. 658–671, May 2007.

[161] B. Liang and Z. J. Haas, "Predictive distance-based mobility management for multidimensional pcs networks," *IEEE/ACM Trans. Netw.*, vol. 11, pp. 718–732, Oct. 2003.

[162] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 483–502, 2002.

[163] K. Lee, S. Hong, S. J. Kim, I. Rhee, and S. Chong, "Slaw: Self-similar least-action human walk," *IEEE/ACM Trans. Netw.*, vol. 20, pp. 515–529, Apr. 2012.

[164] W. Peng and X.-C. Lu, "On the reduction of broadcast redundancy in mobile ad hoc networks," in *Proceedings of the 1st ACM International Symposium on Mobile Ad Hoc Networking & Computing*, MobiHoc '00, (Piscataway, NJ, USA), pp. 129–130, IEEE Press, 2000.

[165] F. Ferrari, M. Zimmerling, L. Mottola, and L. Thiele, "Low-power wireless bus," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, SenSys '12, (New York, NY, USA), pp. 1–14, ACM, 2012.

[166] A. Gaba, S. Voulgaris, K. Iwanicki, and M. van Steen, "Revisiting gossip-based ad-hoc routing," in *Computer Communications and Networks (ICCCN), 2012 21st International Conference on*, pp. 1–6, July 2012.

[167] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying for flooding broadcast messages in mobile wireless networks," in *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 9 - Volume 9*, HICSS '02, (Washington, DC, USA), pp. 298–, IEEE Computer Society, 2002.

[168] W. Lou and J. Wu, "Double-covered broadcast (dcb): a simple reliable broadcast algorithm in manets," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 3, pp. 2084–2095 vol.3, March 2004.

[169] G. Călinescu, I. Măndoiu, P.-J. Wan, and A. Zelikovsky, "Selecting forwarding neighbors in wireless ad hoc networks," in *Proceedings of the 5th*

International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, DIALM '01, (New York, NY, USA), pp. 34–43, ACM, 2001.

[170] J. Wu and F. Dai, "A generic broadcast protocol in ad hoc networks based on self-pruning," in *Proceedings of the 17th International Symposium on Parallel and Distributed Processing*, IPDPS '03, (Washington, DC, USA), pp. 29.1–, IEEE Computer Society, 2003.

[171] J. Wu and F. Dai, "Broadcasting in ad hoc networks based on self-pruning," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 3, pp. 2240–2250 vol.3, March 2003.

[172] A. Keshavarz-Haddad, V. Ribeiro, and R. Riedi, "Drb and dccb: Efficient and robust dynamic broadcast for ad hoc and sensor networks," in *Sensor, Mesh and Ad Hoc Communications and Networks, 2007. SECON '07. 4th Annual IEEE Communications Society Conference on*, pp. 253–262, June 2007.

[173] R. Misra and C. Mandal, "Minimum connected dominating set using a collaborative cover heuristic for ad hoc sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, pp. 292–302, Mar. 2010.

[174] N. Meghanathan and A. Farago, "On the stability of paths, steiner trees and connected dominating sets in mobile ad hoc networks," *Ad Hoc Netw.*, vol. 6, pp. 744–769, July 2008.

[175] I. Stojmenovic, "Comments and corrections to" dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks"," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 15, no. 11, pp. 1054–1055, 2004.

[176] W. Wu, H. Du, X. Jia, Y. Li, and S. C.-H. Huang, "Minimum connected dominating sets and maximal independent sets in unit disk graphs," *Theor. Comput. Sci.*, vol. 352, pp. 1–7, Mar. 2006.

[177] M. Li, P.-J. Wan, and F. Yao, "Tighter approximation bounds for minimum cds in wireless ad hoc networks," in *Algorithms and Computation*, pp. 699–709, Springer, 2009.

[178] A. Vahdatpour, F. Dabiri, M. Moazeni, and M. Sarrafzadeh, "Theoretical bound and practical analysis of connected dominating set in ad hoc

and sensor networks," in *Proceedings of the 22Nd International Symposium on Distributed Computing*, DISC '08, (Berlin, Heidelberg), pp. 481–495, Springer-Verlag, 2008.

[179] D. Liu and M. Prabhakaran, "On randomized broadcasting and gossiping in radio networks," in *Proceedings of the 8th Annual International Conference on Computing and Combinatorics*, COCOON '02, (London, UK, UK), pp. 340–349, Springer-Verlag, 2002.

[180] D. Alistarh, S. Gilbert, R. Guerraoui, and M. Zadimoghaddam, "How efficient can gossip be? (on the cost of resilient information exchange)," in *Proceedings of the 37th International Colloquium Conference on Automata, Languages and Programming: Part II*, ICALP'10, (Berlin, Heidelberg), pp. 115–126, Springer-Verlag, 2010.

[181] N. Alon, A. Bar-Noy, N. Linial, and D. Peleg, "A lower bound for radio broadcast," *J. Comput. Syst. Sci.*, vol. 43, pp. 290–298, Oct. 1991.

[182] H. Xu, L. Huang, C. Qiao, Y. Zhang, and Q. Sun, "Bandwidth-power aware cooperative multipath routing for wireless multimedia sensor networks," *Wireless Communications, IEEE Transactions on*, vol. 11, pp. 1532–1543, April 2012.

[183] S. Kwon and N. B. Shroff, "Analysis of shortest path routing for large multi-hop wireless networks," *IEEE/ACM Trans. Netw.*, vol. 17, pp. 857–869, June 2009.

[184] D. Huang, T. Allen, W. Notz, and N. Zeng, "Global optimization of stochastic black-box systems via sequential kriging meta-models," *Journal of Global Optimization*, vol. 34, no. 3, pp. 441–466, 2006.

[185] J. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *Automatic Control, IEEE Transactions on*, vol. 37, pp. 332–341, Mar 1992.

[186] Y. Seppl, "A chance-constrained programming algorithm," *BIT Numerical Mathematics*, vol. 12, no. 3, pp. 376–399, 1972.

[187] A. Prkopa, "On probabilistic constrained programming," in *Proceedings of the Princeton Symposium on Mathematical Programming*, pp. 113–138, Princeton University Press, 1970.

[188] C. Kam, S. Kompella, G. Nguyen, A. Ephremides, and Z. Jiang, “Frequency selection and relay placement for energy efficiency in underwater acoustic networks,” *Oceanic Engineering, IEEE Journal of*, vol. 39, pp. 331–342, April 2014.

[189] L. Freitag and M. Stojanovic, “Mmse acquisition of dsss acoustic communications signals,” in *OCEANS '04. MTTS/IEEE TECHNO-OCEAN '04*, vol. 1, pp. 14–19 Vol.1, Nov 2004.

[190] M. Gastpar and M. Vetterli, “Source-channel communication in sensor networks,” in *Proceedings of the 2nd International Conference on Information Processing in Sensor Networks*, IPSN'03, (Berlin, Heidelberg), pp. 162–177, Springer-Verlag, 2003.

[191] L. F. Tóth, “Über die dichteste Kugellagerung,” *Math. Z.*, vol. 48, p. 676, 1943.