Title

Cross-Layer Design for Wireless Networks Using Antenna Arrays

Author

Gelal, Ece

Publication Date

2009

UNIVERSITY OF CALIFORNIA
RIVERSIDE

Cross-Layer Design for Wireless Networks Using Antenna Arrays

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy
in
Computer Science
by
Ece Gelal
August 2009

Dissertation Committee:
Professor Srikanth V. Krishnamurthy, Chairperson
Professor Michalis Faloutsos
Professor Neal E. Young

The Dissertation of Ece Gelal is approved by:

Committee Chairperson

University of California, Riverside

ACKNOWLEDGMENTS

I would like to thank the people who have supported me until and during my PhD. First and foremost, I would like to thank my advisor, Prof. Srikanth Krishnamurthy, for his guidance, patience, support, and constant interest in my academic growth. I have learned a lot from his broad knowledge and his unceasing inquisitiveness. I would like to thank Prof. Michalis Faloutsos and Prof. Neal Young for the inspiring discussions and their constructive suggestions that helped me improve my work. I also would like to thank Prof. Ananthanarayanan Chockalingam and Prof. Bhaskar Rao, for their valuable input on the physical layer of MIMO systems. I would like to thank my professors in Sabanci University, Prof. Ozgur Ercetin and Prof. Ozgur Gurbuz, for motivating me to pursue graduate studies. I would like to especially thank Prof. Kemal Oflazer for encouraging me to study computer science.

I had the privilege of interacting with bright people in the Networking Lab in UCR. I am thankful to Doxa, Anirban, Yannis, Tae-Seok, Marios, Kostas, Mustafa for their creative input and feedback at different stages of my work. My collaborations and discussions with them were both educational and fun. My friends outside of lab have been a family to me in Riverside; I am grateful to Anna, Tatev, Tassos, Sherry, Makbule for supporting me during the past five years. I am thankful to my friends overseas, Ozlem, Tugrul, Umut, Yavuz, Efe, Utku, Serkan, who have managed to motivate me from far away. I am most thankful to my parents and my brother, for never stopping to believe in me and for always being there when I need them.

ABSTRACT OF THE DISSERTATION

Cross-Layer Design for Wireless Networks Using Antenna Arrays
by
Ece Gelal

Doctor of Philosophy, Graduate Program in Computer Science
University of California, Riverside, August 2009
Professor Srikanth V. Krishnamurthy, Chairperson

One of the major problems experienced by wireless multi-hop networks is the intermittent network connectivity, which is a consequence of fluctuating link quality due to signal fading. Antenna array technology has been proposed to alleviate the problem of signal fading, and it provides significant performance increase on a single link. However, translating this link-level performance increase to an end-to-end gain in multi-hop networks is not straightforward; a cross-layer design is necessary to efficiently facilitate this translation. In this dissertation, we present cross-layer design approaches for providing end-to-end performance increase in multi-hop networks using antenna arrays. Each approach is designed to utilize a special capability with antenna arrays.

Using antenna arrays, nodes can increase the signal strength in a specific direction; *i.e.,* perform *directional communications.* Using directional communications in a multi-hop network requires nodes to periodically update the directions of their neighbors, which introduces an overhead. We propose topology control algorithms that enable the use of directional communications in multi-hop networks with bounded

overhead. The bounds provided by our Low Degree Spanner (LDS) and Distributed LDS (D-LDS) algorithms are *near-optimal*.

Space-Time Block Coding (STBC) with antenna arrays (referred to as MIMO-STBC) offers significant robustness to fading without an overhead at the higher layers. Robust MIMO-STBC links can also provide performance improvements at the higher layers by the design of proper protocols. Such a design necessitates an accurate representation of the MIMO-STBC link behavior. To date, simplistic representations have been used. We design an accurate representation of MIMO-STBC communications, which we show to have a high fidelity to the MIMO-STBC communications in practice.

Antenna arrays also facilitate the spatial *multiplexing* of signals, allowing a node to transmit and receive multiple signals simultaneously. In a multi-hop network, spatial multiplexing enables receptions from multiple concurrent transmitters. However, such a reception is successful only if both the number and the strength of concurrent transmissions is controlled by a higher-layer mechanism. We design topology control algorithms for activating a maximal number of communications simultaneously, while ensuring that every communication is successful with high probability.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The primary problem in wireless communications is signal fading with propagation distance and due to the environment. Antenna arrays are introduced to alleviate signal degradation due to both of these causes. An array of antenna elements on the transmitter and/or the receiver of a wireless communication allows these nodes to increase the robustness of their communication or to create the perception of multiple independent channels among each other. As a result, antenna arrays significantly enhance the wireless communication quality between a node pair. However, whether the gains on isolated links can also provide an end-to-end performance at the higher layers is unclear.

The goal of the work presented in this dissertation is to propose approaches for translating the link-level performance improvement due to antenna arrays to an end-to-end performance improvement in the multi-hop network. This translation is not a straightforward link-to-network mapping of the performance increase, and it is only possible via a *cross-layer* design. A cross-layer design necessitates a thorough examination of the communication characteristics, and taking these observations into consideration while designing protocols at the higher layers. Consequently, the physical layer is efficiently (and correctly) utilized by the decisions at the higher layers, and end-to-end performance benefits are achieved.

1.1 Directional Communications with Antenna Arrays

One potential method for utilizing antenna arrays on a node is to tune the antenna array such that the transmissions or receptions in a specific direction yield a high signal strength. This method is called *directional beamforming*. Though nodes can benefit from directional transmissions or directional receptions in isolation, in order to avoid asymmetry and to fully utilize the antenna array benefits, it is desirable to use both. *Fully directional* communications are thus formed when both the transmitter and the receiver beamform towards each other's direction; the benefits of fully directional communications have been explored in prior research efforts [48, 42].

A multi-hop network benefits from fully directional communications with a cost of overhead at the higher layers. This is because, in such networks, maintaining connectivity requires nodes to periodically exchange beacons for updating each other's direction information. On the other hand, the overhead in the network can be reduced if each node exchanges these beacons with only a subset of its neighbors. This approach, however, increases the end-to-end path lengths in a multi-hop network (*i.e.,* causes a path stretch). In the multi-hop network, the number of neighbors of a node (*i.e.,* its degree) and the path stretch are inversely related, furthermore, it is resired to have low measures on both of these metrics. The *Low Degree Spanner (LDS)* and *Distributed Low Degree Spanner (D-LDS)* algorithms we propose in Chapter 3 have a near-optimal performance in terms of addressing the tradeoff between node degree and path stretch. Our third algorithm, *Angular Topology Control with Directional Communications (Di-ATC)* uses the intuitions provided by LDS and provides a simpler solution that is conducive to practical implementation in a more connectivity-challenged setting.

1.2 MIMO Space-Time Diversity with Antenna Arrays

A Multiple-Input Multiple-output (MIMO) system is created with antenna arrays due to the fact that each different transmitting antenna-receiving antenna pair yields a different propagation path and therefore a different channel gain. This property can be exploited by *Space-Time Block Coding* (STBC) of the data at the transmitter, using which, space-time encoded copies of a data signal propagate over these "different" channels. The receiver of the MIMO-STBC system can decode the original data even when some of these channels cause significant signal degradation. As a result, MIMO-STBC links are significantly more robust to fluctuating channel gains (compared to the communications with single antenna). Robust MIMO-STBC communications can connect node pairs that have intermittent (or no) connectivity when they use a single antenna.

The benefits on a MIMO-STBC link can translate to a performance improvement at the higher layers, if the higher layer mechanisms build on an accurate model of the MIMO-STBC system at the physical layer. In order to evaluate the accuracy of the models used to date, we study the behavior of these communications on a testbed wherein nodes are equipped with antenna arrays. In Chapter 4, we contrast the link behavior of various MIMO-STBC models to the link behavior in practice.

We point out the essential components of the MIMO-STBC system that need to be considered by an accurate physical layer model. Subsequently, we design a new model that incorporates these components. In Chapter 5 we evaluate the higher-layer performance using each of these models in multi-hop networks. In particular, we contrast the impact of each MIMO-STBC model on the end-to-end performance in multi-hop networks. Our study points out that inaccurate physical layer models of MIMO-STBC communications can result in inaccurate assessments regarding the higher layer performance. We thus emphasize that an accurate estimation of the end-to-end throughput hinges on a realistic characterization of the physical layer. To the best of our knowledge, our work is the first to provide an understanding of how the components of a MIMO-STBC communication interacts with the different layers of the protocol stack.

1.3 MIMO Spatial Multiplexing with Antenna Arrays

Antenna arrays also facilitate the *spatial multiplexing* of multiple signals, which allows a node to transmit and receive multiple signals simultaneously. In a multi-hop network, spatial multiplexing enables nodes to receive from (transmit to) multiple concurrent transmitters (receivers) in their neighborhood. Concurrent reception from multiple transmitters is called *Multi-User MIMO (MU-MIMO) receptions*; similarly, transmission of multiple signals to multiple receivers is called *Multi-User MIMO transmissions*. In a multi-hop network, MU-MIMO receptions are more practical compared to MU-MIMO transmissions, because the latter necessitates that all receivers feed-back the channel gain information to the transmitters. This feed-back has to be frequently updated as the channel is dynamic in wireless communications. Multi-user MIMO receptions are facilitated by Successive Interference Cancellation (SIC). There are two challenges that arise in harnessing the capacity benefits with SIC in a multi-hop network. First, only up to a certain maximum number of concurrent transmissions are allowed. This number is equal to $A - 1$ where A is the number of antenna elements at the receiver. Second, the decoding in each successive iteration is successful only when signals are received with disparate power levels [94, 10].

In Chapter 6 we formulate the problem of *Multi-User MIMO receptions using Successful Interference Cancellation (MU-SIC)* in multi-hop networks. A solution to MU-SIC must identify the communications that can be concurrently active, such that SIC-based decoding is successful at all receivers with an arbitrarily small error probability. We refer to the set of communications that satisfy these criteria to be a "feasible sub-topology". There is a trade-off between the number of feasible sub-topologies and the probability of successful reception on a link in any sub-topology. A link obtains more frequent medium access if fewer sub-topologies are formed; this is possible if each of these sub-topologies include a large number of links, which, however, may reduce the probability of successful decoding in each such sub-topology. MU-SIC is the first formulation that addresses this trade-off towards enabling multi-user MIMO receptions. Finding the optimum solution to the MU-SIC problem, *i.e.,* finding the minimum number of feasible sub-topologies is NP-hard. In Chapter 7 we propose centralized *C-MUSIC* and distributed *D-MUSIC* topology control algorithms, that identify a minimal number of feasible sub-topologies.

In the next chapter we take a closer look at the three physical layer systems using antenna arrays that are studied in this dissertation; we also present previous research efforts related to each system.

Chapter 2

Background on Antenna Arrays

The earliest ideas on the antenna array technology dates back to 1980s, when the initial applications using directional beamforming were proposed [14]. In the subsequent decades, the link-level benefits using various methods with antenna arrays have been investigated at the physical layer. Investigation of the higher-layer performance benefits using this technology did not commence until the twenty-first century. However, within the past decade, numerous protocols have been proposed for translating the physical-layer benefits offered by the antenna array technology to end-to-end performance increase in multi-hop networks. In this chapter, we underline the link-level enhancements that are achievable with antenna arrays, and we briefly discuss prior research efforts that utilize these benefits to obtain higher-layer performance improvements.

2.1 Directional Beamforming with Antenna Arrays

Directional beamforming is achieved by properly tuning the antenna array elements such that the signal strength in a given direction is maximized and/or the power emitted in the unwanted directions is minimized. While the former ability offers a significant enhancement in communication range, the latter ability significantly reduces the interference received from (or transmitted to) unwanted directions and thus, increases spatial reuse.

Directional beamforming antenna arrays (i.e., *directional antennas*) can be used for transmissions and/or receptions (respectively referred to as directional transmissions and directional receptions). In order to fully exploit the potential spatial reuse and range offered by directional antennas, as well as to avoid problems due to asymmetry, it is desirable to use *both* directional transmissions and directional receptions [48, 42], *i.e., to use fully-directional communications.*

2.1.1 Fully-Directional Neighborhood

The primary challenge associated with the use of fully-directional communications in multi-hop networks is the problem of identifying the directional neighbors[1] of each node. Many routing protocols require that at least one-hop neighborhood information is available at the nodes. Omni-directional one-hop broadcast of HELLO messages are no longer accurate in obtaining the directional neighborhood information, therefore novel neighbor discovery and tracking mechanisms have been proposed [42, 95, 96]. Towsley et al., study fully-directional neighbor discovery in static ad hoc networks

[1] Directional neighborhood of a node consists of a node's neighbors it can communicate fully directionally.

[96], and propose a slotted, synchronous scheme called the “Direct discovery”. This work points out that the synchronization requirement can only be obviated via the integration of omni-directional receptions; along this line, the authors also proposed an asynchronous indirect scheme called the “Gossip-based discovery”. Korakis et al., proposed circular sweeping of directional transmissions of HELLO messages, as each node transmits beacons sequentially in every direction and announces its presence [48]. This work, however, requires the reception of the circularly transmitted beacons omni-directionally, providing a smaller neighborhood than what is actually possible.

The second challenge with fully directional communications is that the mobile neighbors of a node are likely to move out of its *angular range* easily, especially when narrow directional beams are used. To avoid losing and re-discovering directional neighbors (which is bandwidth and energy consuming), nodes need to *maintain* their neighbors by periodically updating their locations. In [42], the authors proposed that nodes should periodically *poll* each other to cope with this effect. The periodic polling (*i.e.,* sending/receiving HELLO messages) ensures that every node tracks the motion of its neighbors, and hence preserves connectivity with each neighbor until the neighbor moves out of its *radial range.* By thus proactively maintaining its neighbors' position information, a node obviates the need to rediscover a particular neighbor when it needs to communicate with this neighbor. Neighbor tracking eliminates possible delays and the overhead of reactive neighbor discovery [42][96].

2.1.2 Previous Efforts That Study Directional Beamforming

Literature on Directional Antennas: The development of antenna arrays have inspired the design of novel MAC and routing protocols [21, 53, 46, 48, 68, 79, 91, 105]. A MAC protocol for use with directional antennas must help to combat the problems of **asymmetry in range** and **deafness**. Ramanathan et al., showed that *asymmetry in range can be prevented by allowing only fully directional communications* (of control messages as well as data) [74]. [20] and [60] proposed methods to combat deafness (a node attempting to communicate with a node that is itself in the process of transmission). In [42] deafness is avoided via scheduled fully directional communications. The work describes why *resorting to omnidirectional transmissions or receptions degrades possible spatial reuse, reduces the possible extension in range, and causes asymmetry.*

In Chapter 3 of this dissertation, we focus on bounding the overhead of neighborhood maintenance using fully-directional communications. As we have briefly discussed in the Introduction of this dissertation, bounding this overhead translates to reducing the node degree, which has an adverse impact on end-to-end route lengths. In other words, there is a tradeoff between the sparseness of a topology and the lengths of the paths it contains. In the rest of this section, we discuss related **topology control** efforts that try to minimize *either one* or *both* of these metrics. We first focus on the topology control algorithms that bound the node degree or path stretch; next, we visit the algorithms that try to bound both of these metrics by addressing their tradeoff. We then discuss related work that study the same problem in multi-hop networks that utilize the benefits of antenna arrays.

Topology Control Algorithms that Provide a Low Degree Bound: An initial work addressing the tradeoff between node degree and network connectivity is carried out in [44]. Kleinrock et al. analyze the tradeoff between transmission radius and effective bandwidth at the nodes, and proposed an "optimum" degree bound of 6 for maintaining network connectivity. Sparseness and *bounded degree* have been studied in several other early efforts [11], [12], [25], [80].

More recently, Bose et al. proposed an algorithm to find a bounded-degree Euclidean t-spanner of the *complete* input graph where $t \approx 10.02$; the degree bound is 27 [16]. These centralized algorithms are not practical for ad hoc deployments as they assume that all nodes can directly communicate with each other.

Several distributed algorithms defined proximity graphs of the given UDGs, and utilized the sparseness of these structures for guaranteeing degree bounds. However, although efficient in terms of a localized construction of a sparse topology, bounded-degree proximity structures are shown to yield $O(n)$ path stretch in terms of Euclidean and topological distances (i.e., hop count) when the node distribution is not uniform [15, 25]. It is shown in [57] that no proximity graph with a constant degree bound contains the shortest (in terms of Euclidean distance) path for all node pairs. This work also shows that for $O(1)$-spanner proximity structures (e.g. Yao Graph), the degree bound can be $O(n)$. Given this, the authors use a *Yao-Sink* structure to form a bounded-degree power spanner[2], in which the degree bound can be made tighter by relaxing the spanning ratio (i.e., by changing the parameter k, see Table 2.1). Using the Yao-Sink graph, [58] proposes a localized algorithm to construct *directional* Euclidean spanners having a maximum in-degree of 63. A more recent Euclidean spanner algorithm achieves a tighter degree bound that is strictly greater than 8 [85]; Li et al. improved this bound to 8 in [56].

To date, the best theoretical bound on the node degree (that can be achieved with a distributed approach) is 6 and it is provided by the algorithms in [54, 59, 101] (see Table 2.1). These approaches do not bound the hop stretch.

Topology Control Algorithms that Provide a Low Hop Stretch: Many of the algorithms cited above consider degree bound in conjunction with the path stretch. In these efforts, the goal is to limit the increase in Euclidean length which

[2] A power spanner yields paths on which the energy consumption is at most a constant times larger compared to minimum-energy paths in the UDG [57].

Algorithm	Max. Node Degree	Hop-Stretch
[28]	Not bounded	Constant C, $C > 0$
[54]	6	Not bounded
[56]	8	Not bounded
[57]	$(k+1)^2 - 1$, k: number of cones	Not bounded
[59]	6	Not bounded
[85]	*(i)* 12 ; *(ii)* 9	Not bounded
[100]	$19{+}\lceil 2\pi/\alpha \rceil$, $\alpha{\in}(0, \pi/2)$	Not bounded
[101]	6	Not bounded
D-LDS	**6**	$\boldsymbol{O(1 + \log(\Delta))}$

Table 2.1: Comparison of Distributed Topology Control Algorithms In Terms of Key Features of D-LDS.

is a consequence of reducing node degrees. However, a Euclidean spanner is not necessarily a hop spanner. With a Euclidean spanner, the route between a node pair u,v may have many short hops; in the worst case, the route visits all nodes along the straight line between them (no Euclidean stretch but $O(n)$ hop stretch).

In [9], Alzoubi et al., proposed a localized algorithm for building a subgraph with a hop stretch of 3. The algorithm forms a backbone along which the node degree is bounded with very large values (specifically 295 for the *dominator*s and 7384 for the *connector*s); these bounds do *not* hold in the final topology. Gao et al. proposed the first distributed algorithm that constructs a planar $O(1)$-spanner in terms of both Euclidean and topological distance measures [28]. The constructed geometric spanner enables the use of local routing procedures such as geographical routing, even in existence of node mobility; however, the work does not consider the impact on the node degree. In [18], Burkhart et al. propose an algorithm which, *given* a parameter t, constructs a Euclidean t-spanner of the UDG; the authors indicate that these results are extendable to hop spanners with slight modifications. The work does not bound the node degree.

Summary of the Related Literature Using Omni-Directional Commu-

nications: Topology control algorithms that have been discussed to this point were designed for use with omni-directional communications. Most of these schemes analyzed worst case bounds assuming static ad hoc networks, and they were not evaluated under conditions of mobility. Using these algorithms for topology control in MANETs that use directional (or fully directional) communications may not be efficient, since they do not account for the underlying physical model. Next, we describe the related work on directional communications.

Topology Control Using Directional Antennas: There is limited work to date for constructing topologies that improve the higher layer performance when directional antennas are used in an ad hoc network. Existing studies mostly focus on limiting the power consumption for extending network lifetime. In [36], Huang et al., propose an implementation of the "Cone-based topology control algorithm" (which was previously proposed for omni-directional antennas in [52][101]), using sectorized antennas. The idea is to maintain fewer and closer neighbors in different antenna sectors. The algorithm assumes the use of a neighbor discovery procedure consisting of both directional and omnidirectional modes of communication; as mentioned before, this causes asymmetry and limited spatial reuse. In [37] and [38], Huang et al. constrain power consumption and reduce hop count by using multi-beam directional antennas and adjusting the power intensity independently in each direction. In [67], power use is bounded with beam-switching antennas that may have non-uniform gains within the beamwidth. These efforts do not consider the cost of having high node degrees or the effects of mobility in the ad hoc network with fully directional communications.

There are other studies on the use of directional antennas, for instance for increasing capacity in wireless mesh networks (WMNs) [50, 53, 72]. [50] proposes topology control on WMNs by deploying multiple directional antennas at every node and properly orienting these antennas to create multiple low-interference topologies that are connected. Raman et al. present a mesh network implementation wherein each node is equipped with two high-gain directional antennas [72]; their measurements showcase the potential for using directional antennas to provide low-cost rural connectivity. Roy et al., study the minimum-energy broadcasting problem in wireless multi-hop networks using fully directional communications [78]. These approaches do not consider the impact of mobility.

2.2 Space-Time Diversity with Antenna Arrays

Spatial diversity with MIMO is achieved using Space-Time Block Codes (STBC) [8, 39, 92].

2.2.1 MIMO Space-Time Block Codes (STBC)

With STBC, appropriately encoded data blocks are transmitted from the elements of an antenna array at different times. If the fades between pairs of transmit and receive antenna elements are independent, it has been shown that STBC offers lowered BER for a given SNR, in comparison to a SISO system. In a nutshell, with STBC, the information bits are first mapped on to modulation symbols, which are then input to the STBC encoder. The outputs of the STBC encoder are space-time codewords that determine which symbol is transmitted from which antenna and at what times. The received symbols are combined and decoded using the estimates[3] of channel gains between each antenna pair. Use of STBC reduces the variations in the SNR and as

[3] It is implicitly assumed that the receiver has these estimates (referred to as channel state information or CSI). Typically CSI is estimated from the pilot tones that are transmitted from each antenna.

a result, the probability of correct decoding increases. In the rest of this thesis, we refer to space-time diversity or spatial diversity as MIMO diversity.

In our work, we explore the physical layer characteristics of MIMO-STBC communications, on the specific example of *Alamouti Space-Time Codes* [8]. Alamouti codes provide the maximum possible diversity in the 2x2 MIMO system. The encoding matrix $\mathbf{S_2}$ (the subscript 2 indicates that it is for a 2×2 space-time block code) corresponding to Alamouti codes is given by :

$$\mathbf{S_2} = \begin{bmatrix} s_0 & s_1 \\ -s_1^* & s_0^* \end{bmatrix} \tag{2.1}$$

According to these codes, two symbols are transmitted by two antennas over $2T_s$ time units; first antenna the symbols s_0 and $-s_1^*$ in $(0, T_s)$ and $(T_s, 2T_s)$, respectively and the second antenna transmits the symbols s_1 and s_0^*). At the receiver, received symbols are linearly combined (e.g. using maximum ratio combining) and Alamouti codes are decoded using using maximum-likelihood (ML) detection. More details on STBC are found in [39].

2.2.2 Previous Efforts That Study MIMO STBC

Robust MIMO-STBC links can provide performance improvements at the MAC, routing and transport layers, if novel protocols are designed for use at these layers, and these protocols facilitate an efficient use of the physical layer. MIMO diversity helps to increase the communication range and the robustness to degradation. To date, several research efforts explored the potential performance improvement in multi-hop networks, that is achievable by using MIMO-STBC communications.

The design and evaluation of higher-layer protocols in MIMO networks typically employ **a model to represent the PHY layer**; examples are [86, 41, 29]. In [86], the authors design a cross-layer routing protocol that adapts its behavior based on the feedback from the MAC layer. The model used in the design of this protocol characterizes MIMO diversity gain as follows: (i) if nodes are equipped with k antennas, each MIMO hop spans k SISO hops (due to diversity), or, (ii) diversity increases the packet delivery rate on a link by a constant factor. In [41], diversity is created via node cooperation, which is referred to as Virtual Multi-Input Single-Output or VMISO. In this work, Jakllari et al., use a model, which considers the performance gain due to diversity (diversity gain) to be a constant. A similar model is used in [29], where again the diversity gain is fixed. Both of these two efforts assume that MIMO diversity offers a constant increase of the communication range. However, as we show in Chapter 5, this might not always hold in reality.

In [84], Shrivastava et al., perform an extensive measurement-based study with 802.11n (the IEEE MIMO Draft) and quantify the performance improvements with the different options in 802.11n. In terms of space-time diversity, the authors focus on the performance improvements using maximum ratio combining; however, this work does not consider the cross-layer dependencies among the physical layer and the higher layers.

In [93], Toledo et al., examine the tradeoff between using the SDM and STBC MIMO modes, in terms of the TCP performance between two hosts. The work shows a significant improvement of TCP performance with MIMO diversity, due to the fact that the use of more reliable links allows TCP to operate with a larger sliding window. The work does not consider the interactions between MIMO diversity and other physical layer mechanisms (such as different modulation schemes or error correction codes) or their impact on the higher layer performance. Furthermore, the implications of packetization and routing have not been examined.

There are some physical layer efforts towards building packet error estimators for MIMO-OFDM systems. Bjerke *et al.,* estimate the PER based on the reception SNR in MIMO-OFDM systems with spatial multiplexing [26]; however, their models are only validated via simulations. In [32], Daniels *et al.,* perform measurements of signal waveforms to estimate the fading patterns in an indoor wireless setting. These measurements are used for characterizing the channel simulations that compute PER with various coded modulation schemes. They do not perform packet-level simulations.

In Chapters 4 and 5 of this dissertation, we seek for an accurate representation of the physical layer behavior of MIMO diversity; an accurate representation of the physical layer at the higher layers is essential for an accurate evaluation of the higher-layer behavior. To the best of our knowledge, our work is the first one to examine the correctness of MIMO diversity models that have been used to date. Earlier efforts examined the correctness of the SISO PHY layer models that are used in ad hoc network simulations [49, 89, 90]. In [90], Takai *et al.,* examine the impact of PHY layer characteristics on the performance evaluation of ad hoc routing protocols. In [49], Kotz *et al.,* review six ad hoc simulation assumptions and evaluate their fidelity with regards to a real system via outdoor experiments. Both of these studies emphasize that higher layer protocols may differ in performance when designed on different PHY layer models.

2.3 Space Division Multiplexing (SDM) with Antenna Arrays

In Chapters 6 and 7 of this dissertation, we study the **spatial multiplexing** capability using antenna arrays. We design higher-layer algorithms that facilitate the efficient utilization of MIMO spatial multiplexing in multi-user settings. In this section, we provide a brief background on selective diversity at the transmitter and Successive Interference Cancellation at the receiver of spatially multiplexed MIMO communications. Subsequently, we discuss previous related research work.

2.3.1 MIMO Selective Diversity

In multi-user MIMO communications, antenna array selection at the transmitter side is facilitated via a training process, which typically operates as follows. A transmitter X generates an array of A symbols (where A is the number of its antenna elements), and transmits one symbol with each antenna. A node Y that receives these training symbols maintains a moving average of the channel gain for each transmitting antenna element of node X. Node Y then identifies the antenna element that offers the best channel quality, among all elements in the antenna array of X, and feeds this information back to X [35, 13]. After the session establishment, subsequent packet transmissions on the link (X, Y) use the chosen antenna element.

Note that selection diversity incurs much lower transmission overhead than other multi-user MIMO transmission schemes where all antenna elements are simultaneously utilized. To elaborate, if all antenna elements are to be used for transmissions, some sort of transmitter precoding (*e.g. dirty paper coding* [62]) or power allocation scheme (*e.g.* [83]) needs to be applied in order to enable multi-user MIMO transmissions[69]. With this, each antenna element transmits with a specific weight, which in turn is dictated by receiver feedback. However, due to the channel variability, these weights have to be frequently updated. As a consequence, receivers need to feedback channel state reports for each antenna much more frequently, and this expectedly increases the long-term transmission overhead in the network. On the other hand, with selection diversity *an average channel gain* is estimated for each antenna element; this average is typically stable for prolonged periods of time, *i.e.,* in the order of seconds. Hence, the antenna element that offers the highest average gain is used for extended durations, and this makes the transmission of channel state reports much more infrequent and thus, viable.

2.3.2 Successive Interference Cancellation (SIC) with MIMO Systems

SIC allows receivers to extract a desired signal from a received composite signal; the composite signal is formed by a mixing (*i.e.,* spatial multiplexing) of parallel transmissions in the receiver's neighborhood [27, 104]. SIC is an iterative process. In each iteration, the strongest signal is extracted from the remaining composite signal, as long as its SINR is sufficiently high for successful decoding. Once a signal is successfully decoded (*i.e.,* the transmitted data symbol is correctly estimated), its analog waveform is re-generated at the receiver using signal encoding and modulation, and the obtained waveform is subtracted from the residual composite waveform (*i.e.,* the strongest signal is *removed* from the composite signal). The iterations are continued until the signal of interest is extracted.

At the receiver, different decoding methods may be used in each iteration. With each decoding scheme, the receiver maintains a corresponding equalization matrix

Q. Each node updates its locally-maintained Q with antenna weight coefficients, derived from pilot tones that are transmitted by neighbor nodes. Q is determined based on the decoding scheme in use, in order to boost the power of the target signal relative to the power of the remaining signals that are classified as interference. In our work, we use MMSE-decoding in each iteration [97]; thus, the columns of Q are chosen such that the expected estimation error in detecting a particular user's symbol is minimized. With this method (*i.e.,* MMSE-SIC), a receiver with A antenna elements can successfully decode up to A signals transmitted from A distinct nodes, as long as these signals experience spatially uncorrelated fading and it can obtain the estimates of the fading experienced by each transmission.

2.3.3 Previous Efforts That Study MIMO SDM

To the best of our knowledge, our contribution for facilitating Multi-User MIMO (MU-MIMO) communications is the first work to propose a **topology control** framework for facilitating efficient MIMO SIC operations. In the following, we briefly discuss related prior research work; our contribution distinguishes from these efforts in terms of considering selective diversity and SIC, and in terms of evaluating MU-MIMO in multi-hop networks.

Prior Work for Facilitating Multi-user MIMO Communications: Previously proposed methods for utilizing MIMO space division multiplexing (or spatial multiplexing) are designed for activating the communications at specific times, *i.e.,* for scheduling the links in the network. Sundaresan et al., propose link scheduling algorithms that give medium access priority to the links that belong to multiple contention regions [87]. This work demonstrates that the *stream control* method, using which different transmitters utilize different numbers of antenna elements, leads to significant network-wide throughput benefits. In a more recent study, Wang et al. [99] propose a greedy centralized algorithm for performing stream control; this work quantifies the theoretical upper bound on the achievable gain using this method. Chu et al. propose centralized and distributed algorithms to prioritize the scheduling of links based on the QoS requirements imposed by the nodes and the traffic in the network [22]. Unlike in our work, none of these efforts consider selection diversity at the transmitter or the SINR-dependent SIC at the receiver. Most importantly, unlike our work all of the above studies assume that a receiver is able to decode as many interfering signals as the number of the utilized antenna elements.

Space division multiplexing with MIMO systems has also been studied in [51, 24, 61]. An approximate approach for utilizing the V-BLAST architecture in multi-hop networks is proposed in [51]. The method assumes symbol-level synchronization among nodes and that nodes are able to decode up to 18 simultaneous RTS transmissions, which may not be feasible on the SINR model. In [24], two 8x8 MIMO links have been examined in various distribution of inter-node locations. The examination provides SINR-based decision rules for utilizing spatial multiplexing in large scale multi-hop MIMO networks; however, the work does not examine the performance in such networks.

Applications of SIC: Several recent efforts examine performance benefits with SIC; these studies consider SISO networks. Yi et al., [102] show that a particular distribution of received powers is needed for SIC systems to perform well in CDMA networks. Halperin et al., [33] propose a practical design of SIC in a SISO system using the capturing capability of the wireless radio. They prototype a simple version of their design. Using testbed measurements, they demonstrate that SIC maintains fairness and promotes spatial reuse. The capture effect has also been experimentally studied in [81]; this work also examines the impact of time-ordering of concurrent transmitters. These studies do not consider MIMO operations or multi-hop settings; however, SIC decoding is typically more efficient with an antenna array.

Chapter 3

Efficient Use of Directional Antennas in Multi-Hop Networks

With antenna arrays, nodes can tune their antenna elements such that the signal strength is significantly higher in a specific direction compared to other directions. This is called *directional beamforming*. Fully directional communications are formed when both the transmitter and the receiver beamform in each other's direction; these communications have significantly better robustness to fading. The major challenge in using fully-directional communications in a multi-hop network is the need for determining neighbors' directions for beamforming. Nodes can periodically exchange messages with their neighbors to update the information with regards to their directions. Updates are necessary to ensure that communication with the neighbors is feasible when needed; however, they introduce a communication overhead. The overhead can be reduced if each node exchanges messages with only a sub-set of its neighbors; however, this may increase the hop count of paths between nodes, *i.e.,* causes a *hop stretch*. In this chapter, we study the tradeoff between the hop stretch and the choice of the subset of neighbors that each node periodically updates.

3.1 Introduction

In order to fully exploit the benefits of directional beamforming using antenna arrays, it is essential to exclusively use fully directional communications [42, 73, 105]. In addition to enabling better spatial reuse, fully directional communications also alleviate the phenomena of asymmetry in gain that arises when directional transmissions are used in conjunction with omnidirectional receptions [48].

For communicating with the discovered neighbors in a fully directional fashion, nodes need to acquire up-to-date information with regards to the directions of their neighbors. This can be achieved with a periodic exchange of beacons; however, the overhead incurred due to this exchange may limit achievable performance improvements. Reducing the frequency of periodic beacons to bound the overhead is not a viable option; the frequency is dependent on how fast the nodes move or how fast the environment changes, and increasing the period between updates would cause nodes to have inaccurate neighborhood information. Our approach to limit this overhead is by reducing the *node degrees* in the network. In other words, nodes maintain logical connectivity only with a small subset of their discovered neighbors, and reach the rest of their neighbors via multi-hop routes.

Reducing the node degrees in the network can, however, have an adverse effect on the network connectivity and the hop count of paths between node pairs. Short paths are desirable, due to *(i)* maintaining low levels of packet loss (communication links in a multi-hop wireless network are error-prone and the probability of packet loss increases with route length), *(ii)* avoiding high end-to-end delays (nodes compete for channel access at each hop), and *(iii)* coping better with mobility (longer routes are more likely to fail when nodes move). Thus, while performing topology control to bound the node degree, we try to limit the end-to-end path stretch in the network.

Our first contribution, as presented in Section 3.2 of this chapter, is the design of a topology control algorithm that finds the optimal tradeoff between **low node degree** and **low hop stretch**. Towards the design of this algorithm, we assume that nodes discover their neighbors by *circular directional communications* that are implicitly considered reliable [48]; we thus model the ad hoc network as a unit disk graph. We design the **Low Degree Spanner (LDS)** algorithm, which operates in a centralized fashion. Given the unit disk graph $G{=}(V, E)$ representing all feasible links, LDS finds its sparse subgraph $G'{=}(V, E_{G'})$, wherein the maximum node degree is 6, and for each edge (u, v) in E, the vertices u, v are connected by a path of length $\text{hops}_{G'}(u, v){=}$ $O(\text{hops}_G(u, v){+}\log \Delta)$ (where Δ is the maximum degree of G and $\text{hops}_{G'}(u, v)$ denotes the shortest path length from u to v in G'). We call the latter property as *bounded hop stretch*, and the graph satisfying this property as a *hop-spanner*.

As our second contribution, we design the **Distributed Low Degree Spanner (D-LDS)** algorithm, which satisfies all attractive properties of LDS but provides a localized construction. With D-LDS, nodes make independent decisions on the links they maintain, based only on the knowledge of their two-hop neighborhoods. D-LDS operates on arbitrary networks, does not require synchronization among nodes, scales, and converges quickly to a connected topology with the desired properties. To the best of our knowledge, the most closely comparable previous studies only achieve bounds on *one* of the metrics we consider. These approaches consist of *(i)* algorithms that find subgraphs with a constant hop stretch but do not impose a degree bound [28], and *(ii)* algorithms that guarantee a low degree bound but do not examine hop stretch [54, 59, 101].

Third, drawing on the behavioral observations from D-LDS, we design a simpler but more practical scheme, **Di-ATC (Directional Communications with An-**

gular Topology Control). Di-ATC relaxes the assumption that nodes have accurate two-hop neighborhood information; topology control decisions use the one-hop neighborhood information that is available via fully directional neighbor discovery and maintenance mechanisms. Di-ATC bounds node degree while avoiding high hop stretch, by means of allowing nodes to communicate with a subset of their neighbors such that it is possible to angularly separate the directions in which the neighbors are maintained. To the best of our knowledge, Di-ATC is the first topology control algorithm for lightweight fully-directional neighbor maintenance in a *mobile* ad hoc network.

We evaluate the performance of each of our distributed algorithms (D-LDS and Di-ATC) in terms of the node degree, hop stretch, and the incurred communication overhead. Our results indicate that there is a trade-off between the accuracy of nodes' neighborhood knowledge and the performance in terms of our metrics; with more accurate information nodes can form a sparser connected subgraph. Nevertheless, both our algorithms form topologies that satisfy the desired properties, given arbitrary input networks of different node density, area of deployment, and link reliability.

In Section 3.2 of this chapter, we explain our near-optimal topology control algorithms LDS and D-LDS. In Section 3.3 we present Di-ATC and describe its integration with fully directional neighbor discovery and maintenance. We present the performance evaluation of our distributed protocols in various scenarios in Section 3.4, and we summarize the chapter in Section 3.5.

3.2 Near-Optimal Topology Control Solutions

In this chapter we describe the *Low-Degree Spanner (LDS)* and *Distributed Low-Degree Spanner (D-LDS)* topology control algorithms, which address the tradeoff between the **node degree** and the **path stretch** in multi-hop networks. We first describe the network and communication models, on which we design our topology control solutions.

3.2.1 Model

Both the centralized and distributed constructions to be described in Section 3.2.2 assume that nodes have updated information with regards to their directional neighborhood. As we discuss in Section 3.3, there is an inherent challenge in discovering and maintaining the directional neighborhood in a node's fully directional range; nodes need to beamform in each other's direction and reliably exchange beacons [42, 73, 74]. For ease of the analysis in Section 3.2.2, we assume that the beacons are transmitted reliably (we consider the impact of collisions and side lobes when we design Di-ATC in Section 3.3) in all directions. Given this, we model the directional neighborhood of a node to be circular (Fig. 3.1) and thus, the topology created by the communications among nodes in the network to be a unit disk graph (UDG).

Both LDS and D-LDS algorithms function on an *arbitrary* UDG[1] $G = (V, E)$, where a link exists between two nodes *iff* they are located within each other's communication range. Note that with directional antennas the communicating node pair is typically assumed to have a "line of sight" path. Thus, while the maximum distance at which two nodes can successfully communicate is highly dependent on the environmental characteristics and the availability of line of sight, for simplicity of modeling we assume that signal degradation occurs only due to path loss.

In our network model each node has a distinct ID (e.g. MAC address). Our dis-

[1] In addition to being considered in the design of most topology control algorithms to-date [18][56][101], UDG model has also been used for modeling the neighborhood with directional beamforming [48] and with MIMO [87].

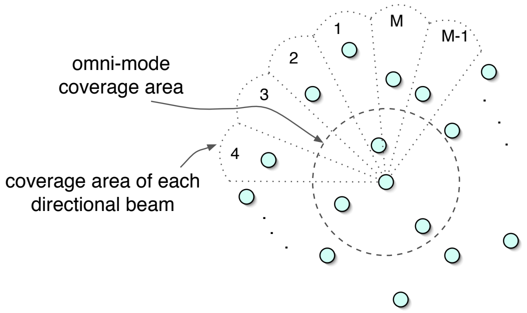


Figure 3.1: Visualizing the circular sweeping of the directional footprint with an M-sector directional antenna.

tributed scheme, D-LDS, assumes that nodes have discovered their *two-hop* neighborhoods[2]; efficient methods for this using omni-directional communications are provided in [19] and [100]. We also assume that nodes are able to infer the relative positions of their neighbors [56, 101]; this can be achieved using the estimation techniques with antenna arrays for computing the Angle-of-Arrival (AOA) [91]. (Alternatively, nodes may be equipped with global positioning system (GPS) units to facilitate this capability. GPS has been used recently in various topology control studies including [54, 55].)

The following two structures are used during the topology construction by both LDS and D-LDS algorithms.

Maximal Independent Set (MIS): Given a graph $G{=}(V, E)$, a subset M of V is an *independent set* if each edge in E is incident on at most one vertex in M. A maximal independent set is an independent set such that for all vertices $v \in (V - M)$ the set $M \cup \{v\}$ is *not* independent; in other words, every vertex that is not in M is adjacent to some vertex in M. It is feasible to construct the set M in $O(n)$ time for UDGs, where $n = |V|$ [63, 88]. Distributed algorithms for finding a maximal independent set

[2] The two-hop neighborhood of node u is defined as the set of nodes N_1 that are neighbors of u, plus those that are neighbors of the nodes in N_1.

are presented in [63, 98].

Balanced Binary Tree: A non-empty binary tree is a *balanced* binary tree, if both its left and right subtrees (T_L and T_R) are balanced binary trees and $|height(T_L) - height(T_R)| \leq 1$. In a balanced binary tree consisting of N nodes, the length of the longest path from the root to a leaf node is $O(\log N)$.

Finally, we introduce the following notation that is used later:
Notation ($\angle(evf)$): When e and f are two edges incident on vertex v, we denote the angle between e and f at v as $\angle(evf)$.

3.2.2 The Low-Degree Spanner (LDS)

Before describing the LDS algorithm, we first show that the bounds achieved with its use are **near-optimal**.

Lemma 3.2.1 *Given a connected unit disk graph G, one cannot always find a connected subgraph G' with a maximum degree of less than five.*

Proof 3.2.2 *Consider the connected UDG with the star topology, where the central node u has 5 independent neighbors, each of which is exactly at a unit distance from u. If we remove the edge between u and* any *neighbor, i.e., make the degree of u less than five, the topology becomes disconnected.*

Lemma 3.2.3 *For a unit disk graph G, any bounded-degree subgraph G' must have* $\text{hops}_{G'}(u,v) = \Omega(\text{hops}_G(u,v) + \log \Delta)$ *for some u, v, where* $\text{hops}_G(u,v)$ *represents the hop count of the shortest path between nodes u, v in G.*

Proof 3.2.4 *Consider any degree-Δ node u and its neighbor set ν in G. In any subgraph $G' \subset G$ with maximum degree δ, let k be the maximum hop-distance from* *u to a node in ν. As there are at most δ^{k+1} nodes within k hops of u, Δ=$|\nu|\leq\delta^{k+1}$, which implies $k\geq\log_\delta(\Delta)-1$.*

The Los Degree Spanner (LDS) Algorithm Description

The construction of a low-degree spanner with LDS consists of three phases: (i) organizing the nodes in *G= (V,E)* into distinct sets, (ii) constructing a backbone that connects these sets, and (iii) assembling the nodes (that are not on the backbone) in each set into balanced binary trees, which are then linked to the backbone.

Phase 1: *Creating groupings on G:* The purpose of this phase is to partition the nodes in V into disjoint groups such that any two nodes that belong to the same group have an edge in E. To achieve this, we define a graph $G_{1/2} = (V, E_{1/2})$ such that there is an edge in $E_{1/2}$ between two nodes in V *iff* they are at most half of the unit distance apart from each other. Then we find a maximal independent set M of $G_{1/2}$ using the algorithm proposed in [88]. Nodes in the set M are called *dominators*; remaining nodes (nodes in $(V - M)$) are called *members.* Each member node is dominated by *one* dominator, which is within a distance of 1/2 from it. The set of member nodes having a common dominator w form *group(w).* Nodes in a group are within a unit distance from each other, i.e., they form a *clique.*

Phase 2: *Construction of the higher-tier backbone:* The purpose of this phase is to construct a bounded-degree backbone H=(V_H, E_H) that interconnects separate groups. To achieve this, LDS: **(a)** defines a set V_H of "backbone nodes", **(b)** selects the edges E_H that span V_H such that the backbone is a *9-spanner* of V_H, with a maximum degree of 6. The decision on each edge is made based on its relative Euclidean length and angle compared to other selected edges that are incident on the considered common node.

Step (a): V_H initially contains all dominators from Phase 1. Two groups are said to be *connected* if there is at least one edge between them in $G = (V, E)$. LDS chooses this edge (if there are multiple such edges, one of them is arbitrarily chosen) and marks the nodes at the end points of this edge as *backbone nodes*; these nodes are added to V_H. New backbone nodes are added to V_H until a pair of backbone nodes is designated for every pair of groups that is connected in G. Finally, in any group that has nonzero member nodes, if the dominator is the only backbone node, an arbitrary additional node s in this group is selected and added to V_H (this is necessary as will be seen later in Phase 3).

Step (b): LDS considers all edges (from G=(V, E)) between vertices in V_H, in the order of nondecreasing length. At each step, the considered edge e=(u, v) is added to E_H *unless* there is an edge f∈ E_H such that f is incident on u or v ***and*** $\angle(evf)$≤ 52°. (f was inserted before e implies that $d(f) \leq d(e)$, where $d(e)$ is the Euclidean length [3] of e.) The phase terminates when all edges between the nodes in V_H have been thus considered.

This construction outputs a Euclidean 9-spanner of the set V_H of backbone nodes. This is facilitated by the choice of backbone edges; the particular angular orientation and length of the selected edges prevents a Euclidean stretch of greater than 9. We formally prove this property in Theorem 3.2.13.

Phase 3: *Connecting the remaining nodes to the higher-tier backbone:* In this phase LDS completes the construction of the connected topology G'=$(V, E_{G'})$ with the desired properties.

a. In every group, a balanced binary tree is formed from the nodes that are not in V_H. **b.** All trees are connected to the backbone such that the maximum degree on

[3] We use the notation *d(e)* for the Euclidean length of edge e and *D(P)* for the Euclidean length of path P.

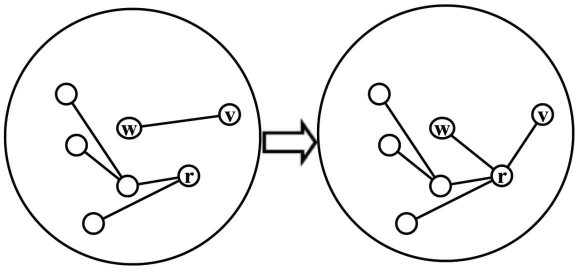


Figure 3.2: Demonstration of how the balanced binary tree is connected to the backbone within the group.

the backbone is preserved.

a. First, in every group that contains member nodes that are not on the backbone, LDS connects them with a rooted balanced binary tree $T(w)$. This construction is possible since every group is a clique. **b. Next,** LDS links the trees to the backbone as follows. Initially, $E_{G'}$ contains the edges in E_H. For each dominator w, if tree $T(w)$ exists (i.e., is not empty), LDS removes an arbitrarily selected edge (w, v) from $E_{G'}$ (Lemma 3.2.7 shows that this edge always exists), and adds two edges (r, w) and (r, v) instead (r is the root of $T(w)$). The edges in $T(w)$ are also added to $E_{G'}$. Figure 3.2 depicts the procedure. In the figure, the edge (w, v) initially belongs to the backbone. The balanced binary tree is attached to the backbone at nodes w and v as discussed above. Note that this process preserves the degrees of w and v.

This phase leads to a hop-spanner as we formally show in Theorem 3.2.19. The hop stretch is bounded in each group by the clique structure, it is also bounded on the backbone due to the properties of Euclidean spanner and MIS.

Next, we show that the topology constructed by LDS has the desired properties in terms of node degree and hop stretch.

3.2.3 Distributed Low-Degree Spanner (D-LDS)

In the following we describe D-LDS, the distributed variant of LDS. With D-LDS, nodes make independent decisions on the links they maintain, based only on local (two-hop neighborhood) information. Using a constant number of broadcasts, D-LDS converges quickly to a connected topology with the near-optimal properties that have been shown for LDS.

D-LDS has four phases.

Phase 1: *Finding the dominators and forming groups:* As with LDS, disjoint groups are formed in this phase. The *dominators* are identified using the distributed algorithm for finding an MIS from [98]. In brief the algorithm works as follows. Initially every node is colored white. A node selected as per some tie-breaking criterion (e.g., the node that has the lowest ID in its one-hop neighborhood) becomes black and declares itself a *dominator*, and it announces its transition with a one-hop broadcast. All of its white neighbors that receive this message become gray and declare themselves to be *dominated* by this dominator. The algorithm terminates when there are no more white nodes. By construction each node has one dominator, since a node changes color only once. Upon termination of this construction, D-LDS defines the set consisting of a dominator and its associated gray nodes ("members") as a *group.* To ensure that all nodes in a group can communicate with each other, *in this phase* nodes reduce their transmission power or increase their receiver sensitivity so that the achievable range is half of the actual maximum range.

Phase 2: *Identification of the backbone nodes:* In this phase, the backbone nodes are identified in a distributed fashion. Given the knowledge of nodes in its two-hop neighborhood, a dominator w designates nodes u, v to be backbone nodes corresponding to *group(w)* and *group(z)*, **if** the following are satisfied:

(i) w has a lower ID than z,

(ii) u,v are in different groups (u∈*group(w)*, v∈*group(z)*, $w{\neq}z$),

(iii) u,v can communicate (*group(w)* and *group(z)* are connected),

(iv) A pair of backbone nodes have not already been assigned for these two groups.

Criterion *(i)* above avoids conflicts in the assignment of backbone node pairs by the corresponding two dominators. The dominator with a larger ID performs the assignment and notifies the other with a single unicast message; no synchronization is needed.

Before proceeding to Phase 3, every dominator w ensures that it is not the only backbone node in its group if *group(w)* contains other nodes; it does so by designating an arbitrary node from *group(w)* as a backbone node. As with LDS, this check ensures that if *group(w)* contains nodes that will be connected to the backbone in Phase 4, then there is at least one link on the backbone such that both its end nodes are in *group(w)*.

Phase 3: *Construction of the Backbone:* In this phase, each backbone node u participates in the distributed construction of a 9-spanner backbone. For this construction, u considers each backbone node v that is its one-hop neighbor, and examines whether the communication link (u, v) is "feasible". The decision regarding the feasibility of a link is made based on the estimated distances to the one-hop backbone neighbors. We define a distance estimation function that facilitates this construction:

Definition (Distance Function): The distance function $\delta : P \rightarrow D$ maps power values to distance values; δ operates as follows:

$$\begin{aligned} P_r(u,v) > P_r(u,z) &\Longrightarrow \hat{d}(u,v) < \hat{d}(u,z) \\ P_r(u,v) = P_r(u,z) \ \&\& \ ID(v) > ID(z) &\Longrightarrow \hat{d}(u,v) < \hat{d}(u,z) \\ P_r(u,v) < P_r(u,z) &\Longrightarrow \hat{d}(u,v) > \hat{d}(u,z) \end{aligned}$$

For its two neighbors v and z, node u estimates v to be closer than z if packets[4] received from v reflect greater power levels than those from z. Ties are broken based on IDs; the node with the higher ID is deemed closer. We assume that δ strictly increases with increasing received power; this is typically true for channels wherein the major source of degradation is path loss.

Definition (Feasibility of a Link): Let Y denote the set of nodes that are closer to node u than v; nodes in Y reside *inside* the circle centered at u with a radius of $\hat{d}(u,v)$. Similarly, let Z denote the set of nodes that are closer to v than u. Initially all links are unprocessed and unmarked.

1. A link (u,v) is *feasible*, if **both** of the following local conditions are satisfied:

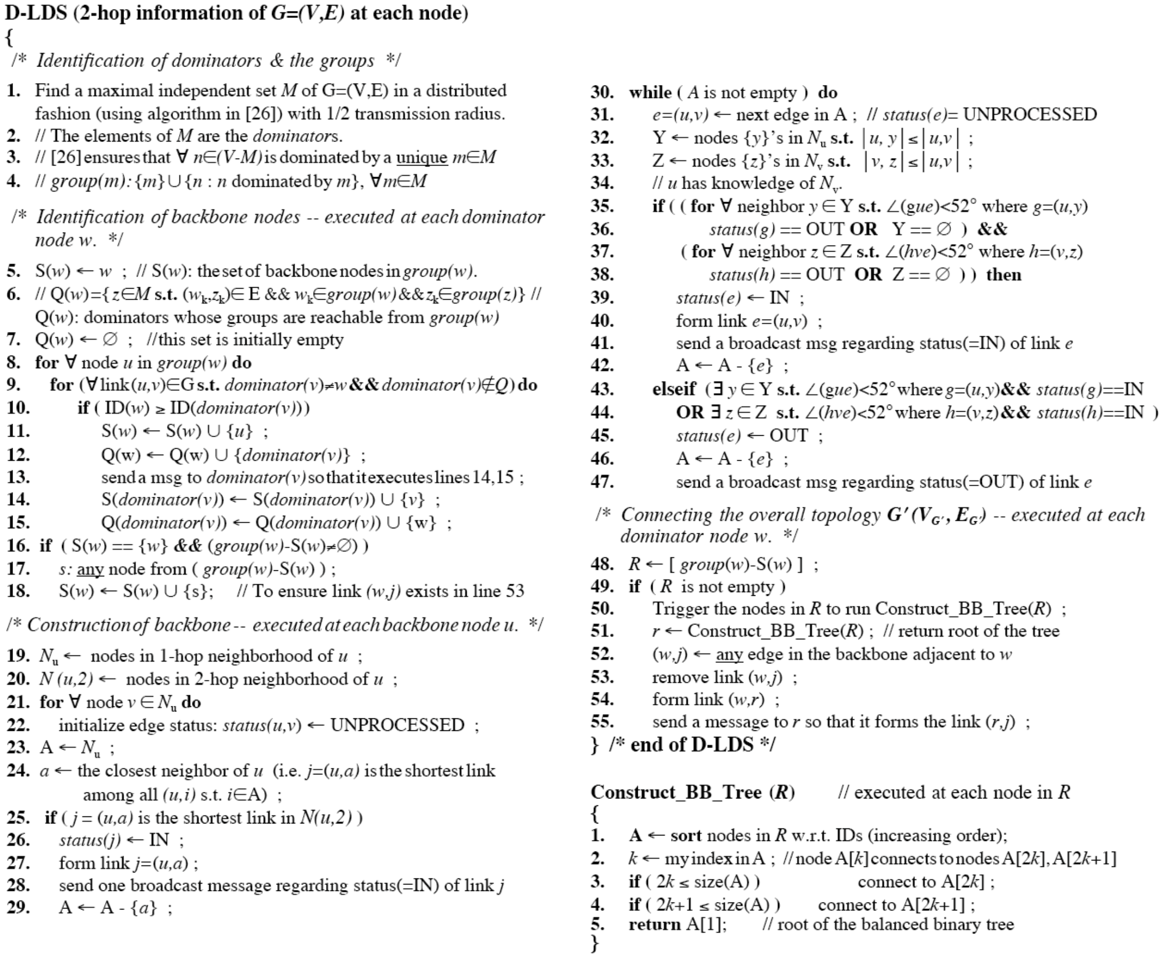
   (i) Every link (u,y_i) $(y_i \in Y)$ that would make an angle of $< 52°$ with (u,v) must be marked as "unfeasible".

   (ii) Every link (v,z_i) $(z_i \in Z)$ that would make an angle of $< 52°$ with (v,u) must be marked as "unfeasible".

   A *feasible* communication link is added to the topology.

2. For link (u,v) to be *unfeasible*, it is necessary and sufficient that **either** of the following two conditions is satisfied:

[4] To prevent distance information from becoming stale, periodic updates will be necessary. We assume that the HELLO messages facilitate this process.



```
D-LDS (2-hop information of $G=(V,E)$ at each node)
{
 /* Identification of dominators & the groups */
1.  Find a maximal independent set $M$ of G=(V,E) in a distributed
    fashion (using algorithm in [26]) with 1/2 transmission radius.
2.  // The elements of $M$ are the dominators.
3.  // [26] ensures that $\forall\, n \in (V-M)$ is dominated by a unique $m \in M$
4.  // $group(m)$: $\{m\} \cup \{n : n$ dominated by $m\}$, $\forall m \in M$

 /* Identification of backbone nodes -- executed at each dominator
    node w. */
5.  S(w) ← w ; // S(w): the set of backbone nodes in group(w).
6.  // $Q(w)=\{z \in M$ s.t. $(w_k,z_k) \in E$ && $w_k \in group(w)$ && $z_k \in group(z)\}$ //
    Q(w): dominators whose groups are reachable from group(w)
7.  Q(w) ← ∅ ;  //this set is initially empty
8.  for ∀ node u in group(w) do
9.      for (∀ link $(u,v) \in G$ s.t. $dominator(v) \neq w$ && $dominator(v) \notin Q$) do
10.         if ( ID(w) ≥ ID(dominator(v)) )
11.             S(w) ← S(w) ∪ {u} ;
12.             Q(w) ← Q(w) ∪ {dominator(v)} ;
13.             send a msg to dominator(v) so that it executes lines 14,15 ;
14.             S(dominator(v)) ← S(dominator(v)) ∪ {v} ;
15.             Q(dominator(v)) ← Q(dominator(v)) ∪ {w} ;
16. if ( S(w) == {w} && (group(w)-S(w)≠∅) )
17.     s: any node from ( group(w)-S(w) ) ;
18.     S(w) ← S(w) ∪ {s};   // To ensure link (w,j) exists in line 53

/* Construction of backbone -- executed at each backbone node u. */
19. $N_u$ ← nodes in 1-hop neighborhood of u ;
20. N (u,2) ← nodes in 2-hop neighborhood of u ;
21. for ∀ node $v \in N_u$ do
22.     initialize edge status: status(u,v) ← UNPROCESSED ;
23. A ← $N_u$ ;
24. a ← the closest neighbor of u (i.e. j=(u,a) is the shortest link
            among all (u,i) s.t. $i \in A$) ;
25. if ( j = (u,a) is the shortest link in N(u,2) )
26.     status(j) ← IN ;
27.     form link j=(u,a) ;
28.     send one broadcast message regarding status(=IN) of link j
29.     A ← A - {a} ;
30. while ( A is not empty ) do
31.     e=(u,v) ← next edge in A ; // status(e)= UNPROCESSED
32.     Y ← nodes {y}'s in $N_u$ s.t. $|u, y| \leq |u,v|$ ;
33.     Z ← nodes {z}'s in $N_v$ s.t. $|v, z| \leq |u,v|$ ;
34.     // u has knowledge of $N_v$.
35.     if ( ( for ∀ neighbor $y \in Y$ s.t. $\angle(gue) < 52°$ where g=(u,y)
36.               status(g) == OUT OR  Y == ∅ ) &&
37.          ( for ∀ neighbor $z \in Z$ s.t. $\angle(hve) < 52°$ where h=(v,z)
38.               status(h) == OUT OR Z == ∅ ) ) then
39.         status(e) ← IN ;
40.         form link e=(u,v) ;
41.         send a broadcast msg regarding status(=IN) of link e
42.         A ← A - {e} ;
43.     elseif ( $\exists\, y \in Y$ s.t. $\angle(gue) < 52°$ where g=(u,y) && status(g)==IN
44.         OR $\exists\, z \in Z$ s.t. $\angle(hve) < 52°$ where h=(v,z) && status(h)==IN )
45.         status(e) ← OUT ;
46.         A ← A - {e} ;
47.         send a broadcast msg regarding status(=OUT) of link e
 /* Connecting the overall topology $G'(V_{G'}, E_{G'})$ -- executed at each
    dominator node w. */
48. R ← [ group(w)-S(w) ] ;
49. if ( R is not empty )
50.     Trigger the nodes in R to run Construct_BB_Tree(R) ;
51.     r ← Construct_BB_Tree(R) ; // return root of the tree
52.     (w,j) ← any edge in the backbone adjacent to w
53.     remove link (w,j) ;
54.     form link (w,r) ;
55.     send a message to r so that it forms the link (r,j) ;
} /* end of D-LDS */

Construct_BB_Tree (R)      // executed at each node in R
{
1.  A ← sort nodes in R w.r.t. IDs (increasing order);
2.  k ← my index in A ; // node A[k] connects to nodes A[2k], A[2k+1]
3.  if ( 2k ≤ size(A) )            connect to A[2k] ;
4.  if ( 2k+1 ≤ size(A) )      connect to A[2k+1] ;
5.  return A[1];      // root of the balanced binary tree
}
```

Figure 3.3: Distributed Algorithm D-LDS constructs a low degree hop spanner with $O(1)$ broadcasts at each node.

(i) There is a link (u, y_i) $(y_i \in Y)$ added to the topology, making an angle of less than 52° with (u, v)

(ii) There is a link (v, z_i) $(z_i \in Z)$ added to the topology, making an angle of less than 52° with (u, v).

If link (u, v) is *unfeasible*, both endpoints keep this information; nodes u and v do not communicate in the final topology.

Each backbone node performs a local broadcast (which is in turn propagated to its two-hop neighborhood) upon deciding the status of its link with a backbone neighbor. The links in the topology at the end of this phase constitute the backbone.

Phase 4: *Finalizing the construction of the connected topology:* In every group, nodes that are not backbone nodes (if any) will form a balanced binary tree via a distributed procedure as shown in Figure 3.3. Let the set of such nodes in *group(w)* be $R(w)$. Tree construction in the group is triggered by the dominator w and it is performed concurrently at each node in $R(w)$. Each node carries out the following: ***(i)*** It sorts the nodes in $R(w)$ as per their IDs, in increasing order. (The sorted array is unique and is *the same* at all the nodes in $R(w)$, given that two-hop neighborhood information is available at all nodes.) ***(ii)*** If it is at index k in the sorted order, it connects to nodes at indices $2k$ and $2k + 1$ (if $2k$ or $2k$+1 do not exceed the number of elements in $R(w)$). This construction does not require any message exchange, and the tree is unique for a given set $R(w)$.

Without loss of generality, let node r be the root of the tree constructed in *group(w)*. r connects to the backbone, as w removes link (w, j) (the shortest backbone link adjacent to w by Lemma 3.2.7) and initiates the construction of the links (w, r) and (r, j). This procedure requires two unicast messages: from w to j for the removal of the link (w, j), and from w to r, to trigger the formation of link (r, j). Phase 4 terminates after this construction is complete for all groups.

D-LDS algorithm is depicted in Figure 3.3.

3.2.4 Near-Optimality of LDS and D-LDS

Analysis of LDS

Lemma 3.2.5 *Let* $e{=}(u,v)$*,*$f{=}(u,w)$ *be two edges incident on node* u*, and let* $g{=}(v,w)$ *be the edge across the angle* $\angle(euf)$*. If* $d(f){<}d(e)$ *and* $\angle(euf){\leq}52°$*, then* $d(g){<}d(e)$*.*

Proof 3.2.6 *Assume* $d(g){>}d(e)$*. Then* $\angle(euf)$ *is the largest angle in* $\triangle(uvw)$*, which implies* $\angle(euf){\geq}60°$*. However,* $\angle(euf){\leq}52°$*, which contradicts the assumption; therefore* $d(g){<}d(e)$*.*

Lemma 3.2.7 *For a dominator* u*, if the number of nodes in* group(u) *is greater than one, there is at least another backbone node* v *in* $H{=}(V_H,E_H)$ *such that* $v \in \text{group}(u)$*. Specifically,* $(u,v){\in}E_H$*, where* v *is the backbone node* closest to u*.*

Proof 3.2.8 *We prove the lemma by contradiction. Note that* group(u) *contains at least two nodes in* V_H *by construction (Phase 2). Assume that* $v \in V_H$ *is the closest backbone node to dominator* u[5]*, and that edge* $(u,v) \notin E_H$*. This implies that there is an edge* (u,w) *or* (v,z) *in* E_H *(*w*,*$z \in V_H$*) that* blocks[6] (u,v)*. However, the blocking edge cannot be* (u,w) *because this would imply* $d(u,w) \leq d(u,v)$ *which contradicts the assumption. Therefore, it must be that* $(v,z) \in E_H$ *blocks* (u,v)*. Then* $d(v,z){\leq}d(u,v)$ *and* $\angle(v,z)v(v,u) < 52°$*. By Lemma 3.2.5,* $d(u,z){<}d(u,v)$*, which contradicts the assumption.*

Theorem 3.2.9 *The maximum node degree in* G' *is* 6*.*

Proof 3.2.10 *It suffices to show that* $H(V_H,E_H)$ *has a maximum degree of* 6*, because Phase 3 of the algorithm ensures that in each group:*

[5] In case of ties, the backbone node with smaller ID is deemed closer.

[6] We say that an edge e is "blocked" by an adjacent edge f *iff* $d(f) \leq d(e)$ and $\angle(evf) < 52°$.

(i) member nodes that are not backbone nodes are assembled into balanced binary trees, such that the tree has a maximum node degree of 3*,*

(ii) the degree of the root increases by 2 *when connected to the backbone, but becomes at most* 4*,*

(iii) the degree of any node in H is preserved in constructing G' upon appending the balanced binary trees.

Then, this theorem holds due to the following Lemma.

Lemma 3.2.11 *The degree of any node in H is at most* 6*.*

Proof 3.2.12 *Assume $\exists$ a node u in V_H with degree deg(u)>6. Then, two of the edges that are incident on u* must *create an angle of at most* (360/7)<52°*. However, by construction, no two edges in H make an angle of less than* 52°*.*

Theorem 3.2.13 *Each pair of backbone nodes u, v such that edge $e = (u, v) \in G$, are connected in H by a path P such that $D(P) \leq 9 \cdot d(e)$.*

Proof 3.2.14 *The proof is by induction on the edges between the backbone nodes in G, in the order they are considered (whether to be a backbone edge) in constructing E_H. Consider one such edge e=(u, v). If (u, v)∈E_H, then the theorem holds. If not, then $\exists$ an edge f=(u, w) ∈E_H such that $d(u, w)$≤$d(u, v)$ and $\angle(euf)$<52°. By Lemma 3.2.5 $d(v, w)$<$d(u, v)$ and thus, (v, w) was considered by LDS prior to (u, v). Consider path $P(v, u)$ formed with $P'(v, w)$ followed by edge (w, u). For induction, we postulate that, $\exists$ a path $P'(v, w)$ in E_H such that $d(P')$≤9·$d(v, w)$. We use this to show $d_H(u, v)$≤9·$d(u, v)$. Then, $d(P)$≤9·$d(v, w)$+$d(w, u)$. With this, it is enough to show:*

$$d(u, w) + 9 \cdot d(v, w) \leq 9 \cdot d(u, v). \tag{3.1}$$

Or, equivalently,

$$\frac{d(u,w)}{d(u,v) - d(v,w)} \leq 9. \tag{3.2}$$

Let w' be a point on (u,v) s.t. $d(v,w')$=$d(v,w)$. Thus, the triangle $\triangle(vww')$ is isosceles with equal angles at w and w', as shown in Figure 3.4(a). Then $d(u,w')$=$d(u,v)$-$d(v,w)$, and inequality(3.2) becomes:

$$\frac{d(u,w)}{d(u,w')} \leq 9. \tag{3.3}$$

For any fixed $d(u,w)$ and a fixed angle $\angle(euf)$ at u, $d(u,w')$ is minimized when $d(u,v)$ is minimized. The smallest value $d(u,v)$ can take is $d(u,w)$, since inequality $d(u,w) \leq d(u,v)$ has to be satisfied. Then the value of the left hand side of inequality(3.3) is maximum when the denominator $d(u,w')$ is minimized, i.e., when $d(u,w)$=$d(u,v)$. This case is shown in the Figure 3.4(b). The triangle $\triangle(uvw)$ is isosceles with equal angles at v and w. Let z be the midpoint of (v,w). By considering the right triangle $\triangle(uvz)$ we compute $d(v,z)$ and thereby $d(v,w)$ as:

$$d(v,w) = 2 \cdot d(v,z) = 2 \cdot d(u,v) \cdot \sin(\frac{\angle(euf)}{2}). \tag{3.4}$$

$$\begin{aligned}
\textit{Thus,} \qquad & d(u,w) + 9 \cdot d(v,w) = d(u,v) + 9 \cdot d(v,w) \\
& \leq d(u,v) + 18 \cdot d(u,v) \cdot \sin(\frac{\angle(euf)}{2}) \\
& \leq d(u,v) \cdot \left(1 + 18 \cdot \sin(\frac{\angle(euf)}{2})\right) \\
& \leq 9 \cdot d(u,v) \qquad (3.5)
\end{aligned}$$

The last step in inequality(3.5) is due to $\angle(euf)$<52°, which proves the theorem.

Proposition 3.2.15 *The factor 9 can be improved to 8 by considering angles of*

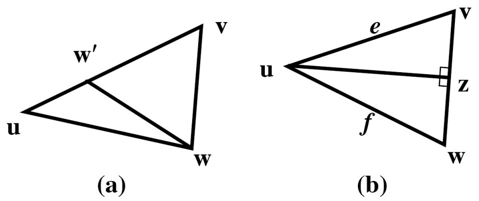


Figure 3.4: Figure to aid the discussion of Theorem 3.2.13.

$51.5 > 360/7$ *degrees above instead of* $52°$.

Lemma 3.2.16 *The number of groups connected to any given group* g *is* $O(1)$.

Proof 3.2.17 *In Phase 1, dominators are chosen such that they are at least* $1/2$ *units apart. Draw disks of radius* $1/4$ *centered at each dominator whose groups are connected to* g*. These disks are disjoint and all of them lie within a circle of radius* 2 *centered at* g*'s dominator. The number of such disks is* $O(1)$.

Corollary 3.2.18 *The number of backbone nodes in each group is* $O(1)$*. Therefore, the number of backbone nodes that are reachable from any backbone node is* $O(1)$.

Theorem 3.2.19 *For any two nodes* u, v *that are connected by a path* P *in* $G = (V, E)$*, there is a path* P' *in* $G' = (V, E_{G'})$*, such that* $\text{hops}_{G'}(P') = O\left(\text{hops}_G(P) + \log \Delta\right)$*. (*$\Delta$ *is the maximum node degree in* G*.)*

Proof 3.2.20 *As each group forms a clique in* G*, each group has at most* $\Delta+1$ *nodes. Consider the path* P *from* u *to* v *in* G*; let* $g_1, g_2, g_3, \ldots, g_{k+1}$ *be the sequence of groups that* P *traverses.* P *has at least* k *edges: for each* $i = 1..k$*, there is an edge from* g_i *to* g_{i+1} *in* G*. By construction, there is such an edge also in* E_H *for each* $i = 1..k$*. Let* a_i *and* b_i *be backbone nodes in* g_i*, such that* a_i *connects* g_i *to* g_{i-1} *and* b_i *connects* g_i *to* g_{i+1}*. This implies* $d(b_{i-1}, a_i) \leq 1$*,* $d(a_i, b_i) \leq 1$ *and* $d(b_i, a_{i+1}) \leq 1$ *in* G*. By*

Theorem 3.2.13 there is a path p_i from b_i to a_{i+1} and a path q_i from a_i to b_i in the backbone graph $H = (V_H, E_H)$, such that $D(p_i)$ and $D(q_i)$ are each ≤ 9. Construct P' from u to v in G' as follows.

From u, traverse the edges in the balanced binary tree of u's group (g_1, w.l.o.g.) to get to the root. From the root get to a_1. Then traverse paths $p_1, q_2, p_2, q_3, \ldots, p_k, q_{k+1}$ to get from a_1 to b_{k+1}. From b_{k+1} get to the root of the balanced binary tree in v's group (g_{k+1}); then traverse the tree to get from b_{k+1} to v. Traversal in each binary tree requires at most $\log_2 \Delta$ edges. As each of the paths p_i and q_i has Euclidean length at most 9, they contain nodes from $O(1)$ groups (as they pass through $O(1)$ groups). By Corollary 3.2.18 each such path has $O(1)$ edges (connecting at most all the backbone nodes in all visited groups). The number of these paths is $2k$. Additionally, the path may need to traverse $O(\log \Delta)$ edges in order to reach the backbone (via the binary balanced trees). Therefore the hop count of path P' is $O(k + \log \Delta)$.

Corollary 3.2.21 *For two nodes u, v that are connected by a path P in $G = (V, E)$, there is a path P' that connects them in $G' = (V, E_{G'})$ such that $D_{G'}(P')$=$O\left(D_G(P) + \log \Delta\right)$.*

Proof 3.2.22 *The proof follows from Theorem 3.2.19 and from fact that each hop is of at most unit length.*

Lemma 3.2.23 *The time complexity of LDS is $O(n \log n)$.*

Proof 3.2.24 *A maximal independent set can be found in $O(n)$ time [63]. By Corollary 3.2.18, the number of edges on the backbone is $O(n)$ (the number of groups being $O(n)$ in the worst case). Sorting these edges takes $O(n \log n)$ time. Constructing each balanced binary tree takes $O(\Delta \text{log} \Delta)$ time (requires sorting the nodes in a group w.r.t. their ID's). Therefore, the time complexity of LDS is $O(n \log n)$.*

Theorem 3.2.25 *The topology constructed by D-LDS has a maximum degree of* 6.

Proof 3.2.26 *The proof follows from that of Theorem 3.2.9 as D-LDS emulates LDS in all phases.*

Theorem 3.2.27 *Let $G' = (V, E_{G'})$ be the topology constructed by D-LDS. For each pair of nodes u, v that were connected by a path P in G, they are connected by path P' in G', such that* $\text{hops}_{G'}(u,v)$=$O(\text{hops}_G(u,v)$+$\log\Delta)$.

Proof 3.2.28 *First we show that the backbone construction in Phase 2 of LDS and in Phase 3 of D-LDS output identical decisions with regards to the edges in the final topology; this is despite D-LDS working with only local information at the nodes. At the end of these phases the constructed backbones are the same, as long as the input set of backbone nodes are the same.*

Recall that LDS sorts all possible edges between nodes in V_H in nondecreasing order, and considers one edge at a time. In D-LDS, a link $e = (u, v)$ is constructed, only if **all** *edges that can* potentially *block it are marked* unfeasible. *This implies that, though the order in which the edges are processed may differ in LDS and D-LDS, the* same *decision is eventually made by both algorithms, regarding whether or not an edge is feasible or unfeasible.*

Thus, by Theorem 3.2.19, G' has bounded hop stretch.

Corollary 3.2.29 *For each pair of nodes u, v that had a path connecting them in G, there is a path P' connecting them in G', such that $D_{G'}(P') = O\left(D_G(P) + \log\Delta\right)$, where $D_G(P)$ denotes the Euclidean length of path P.*

Proof 3.2.30 *The proof follows from Theorem 3.2.19, Corollary 3.2.21 and Theorem 3.2.27.*

Theorem 3.2.31 *For any input graph G, D-LDS constructs the final topology G' using $O(n)$ messages (in the worst case).*

Proof 3.2.32 *Two-hop neighborhood information can be acquired at all nodes with $O(n)$ messages [19]. Message complexity of finding an MIS is $O(n)$ [98]. The number of backbone links is at most $O(n)$ (Lemma 3.2.23); thus, the total number of broadcasts for backbone construction is $O(n)$ (each backbone node performs one local broadcast upon deciding the status of a link). Balanced binary tree construction does not require message exchanges (two-hop neighborhood is already discovered). No broadcasts are necessary for linking the trees to the backbone.*

3.3 Angular Topology Control with Directional Antennas (Di-ATC)

The *near-optimal* (defined in Section 3.2.2) construction with the LDS and D-LDS algorithms are facilitated by modeling the communication network as a unit disk graph. However, although popularly employed by most topology control studies to date, the UDG model does not accurately reflect the communication characteristics with directional antennas. In particular, we assume that the directional antennas have idealized footprints (i.e., no sidelobes are present). More importantly, the operation of D-LDS relies on accurate neighborhood information and a connected topology. In practice, due to the characteristics of the fully directional neighbor discovery process, D-LDS may not perform efficiently in all scenarios.

On the other hand, D-LDS algorithm provides insights that can be utilized in designing a more practical topology control scheme for fully-directional communications. In particular, D-LDS demonstrates that when each node in the network maintains logical connectivity with its neighbors that are angularly separated, the hop stretch in the constructed *sparse* topology can be constrained. Based on the key insights gained from the construction of D-LDS we design a simpler but more practical topology control scheme, Di-ATC, for facilitating fully directional communications with bounded overhead. We first describe the network model used for this design.

3.3.1 Model for the Design of Di-ATC

In the network model for Di-ATC, we employ a beam-steering directional antenna which can target its boresight to an arbitrary direction. Our antenna system functions in two modes. When a node is searching for neighbors (to be discussed in detail in Section 3.3), the antenna functions as a switched beam antenna. It uses a fixed beamwidth Θ and can scan $2\pi/\Theta$ fixed directions. A node can either transmit or receive directionally at a given time. Once a node establishes a connection with a neighbor, it communicates with this neighbor by beamforming in its direction. Nodes continuously monitor the DOA (direction of arrival) or the AOA of their neighbors, from the signals they receive. A node updates its antenna weighting coefficients to point its main lobe towards the target node. We assume that the time taken by an antenna to adapt its weighting coefficients is negligible in comparison to the duration of a frame exchange.

3.3.2 Di-ATC Specification

With Di-ATC, in addition to bounding the node degree without causing large hop stretch, we conform to the following design goals.

- *First,* it is essential that topology control is tightly knit with fully directional neighbor discovery and maintenance mechanisms. Omnidirectional transmissions or receptions must not be invoked in any phase of the algorithm.

- *Second,* topology control must preserve network connectivity. In particular, with fully directional neighbor discovery (as will be described later in this section), nodes may not acquire accurate information with regards to their neighbors; aggressive topology control decisions (for low degree) may lead to network partitions or isolation.

- *Third,* since it is expensive (in terms of power, bandwidth, and time) to exchange directional control messages with neighbors, topology control must operate in a decentralized fashion and using information that is local and limited to the extent possible. Topology control can consequently scale with network size and in the presence of node mobility.

With Di-ATC, given a degree bound K, nodes compute the necessary angular separation that must be maintained among their neighbors in the formed topology. Nodes then communicate with only a subset of the discovered neighbors; nodes track (i.e., exchange update beacons with) these neighbors periodically. Thus, nodes proactively maintain the information in terms of 'in which direction to beamform' in order to communicate with a neighbor. The decision at a node u, with regards to which neighbors it will track, is made based on its current degree, the current degrees and the angular positions of the neighbors that it is tracking (periodic messages should include the degree of the sender), and the degree bound K.

Given K, u decides to maintain a link with a discovered node v if *any* of the following conditions hold:

1. its current node degree ($deg(u)$) is less than K, or

2. $deg(u)$=K, but u does not have any other neighbors within $\Theta = \lceil 360/(K+1) \rceil^\circ$ of the link (u, v) (If this case occurs, u has at least two other neighbors that are angularly apart by less than Θ. u will remove one of these links in favor of adding the link (u, v).), or

3. $deg(u) = K$ and u has a neighbor z such that the angle between links (u, z) and (u, v) is less than or equal to Θ, but $deg(v)$ is strictly smaller than $deg(z)$. (In this case u will attempt the removal of link (u, z) in favor of adding the link (u, v).)

The first condition implies that a node will maintain a discovered node as long has it has fewer neighbors than the degree bound. This behavior distinguishes Di-ATC from D-LDS. The condition is imposed for increasing the probability of maintaining connectivity; the decision helps to constrain the hop stretch as well. The second and third conditions are motivated by our design requirement of angularly separating the directions in which the neighbors are maintained. These criteria allow nodes to reach spatially separated areas with low hop stretch. A node u determines whether it has a conflict (based on the three criteria above) while adding the discovered neighbor v to its neighbor set; link (u, v) is formed only if *both* u and v agree on tracking each other independently, and reliably exchange topology control messages.

Note that these decisions do not rely on perfect directional neighborhood information or perfect channel conditions. These heuristics are designed to support network connectivity; this is essential because nodes may easily lose their neighbors with mobility when omni directional communications are not invoked.

3.3.3 Integrating Di-ATC with Neighbor Discovery & Maintenance

We propose an integrated framework that allows seamless interactions among three functionalities; fully directional neighbor discovery, topology control on the discovered nodes, and directional maintenance of the neighbors that are chosen after topology control. As discussed earlier, the framework requires that nodes only *track* a subset of their directional neighbors. Di-ATC tries to find, given the available neighborhood information, the subset that offers good connectivity and short routes.

Our framework executes in cycles, as depicted in the state diagram in Figure 3.5. All three of the aforementioned functionalities are invoked by each node in the network, in every cycle. Nodes are synchronized in terms of time slots that constitute these cycles. Synchronization is an essential characteristic for neighbor discovery and communications using fully directional communications [42, 74, 96]. Specifically, to facilitate a fully directional link, both end nodes must beamform towards each other at the same time. Synchronization can be achieved via solutions such as the method proposed in [77]; guard bands could also be employed to facilitate synchronization. We wish to point out that (a) only a coarse-grained synchronization at the frame level is needed, (b) only local synchronization among neighbors is needed, and (c) synchronization is a requirement of fully directional communications and not of our techniques in particular.

For fully directional neighbor discovery, each node beamforms towards a randomly chosen direction (or antenna sector, as shown in Fig. 3.1) at specific time-slots, and either transmits a HELLO packet in this direction or listens in the anticipation of receiving one. If a successful communication occurs, the node to first receive the HELLO beacon responds with one of its own to complete the handshake; the successful handshake implies that the node pair under discussion have discovered each other.

Directional neighbor discovery is a continuous process; nodes discover new neighbors as the topology changes due to mobility or due to variations in the environment. As shown in Figure 3.5, in each cycle several time-slots need to be allocated for the directional search for new neighbors. In particular, due to the probabilistic nature of the fully directional neighbor discovery process, it takes many cycles until nodes discover a fair percentage of their neighbors [42][74][96]. In our framework we use Di-ATC with the neighbor discovery protocol proposed in [42].

We remark that other methods have been proposed for directional neighbor discovery; these approaches either include circular transmissions [48], or select a number of fixed beams [82], or apply transmit-only beamforming [74]. With all these schemes, the receptions during neighbor discovery are omni-directional. As a consequence, these approaches may provide a smaller neighborhood than what is actually achievable with fully directional communications. These approaches also are prone to inefficiencies that arise due to asymmetry in range.

Di-ATC is invoked at both nodes (u and v) that discover each other. In the following time slot, the node pair exchange the outcome of their Di-ATC execution. The new link (u, v) is established if both u and v *want* this link (based on the three criteria listed above) *and* can reliably exchange messages to agree on the link construction. Such local and dynamic decision-making renders the framework scalable under varying network conditions.

Tracking discovered neighbors is crucial with fully directional communications, especially when the nodes are mobile or when the environment changes dynamically. In our framework, nodes proceed to the neighbor tracking phase after

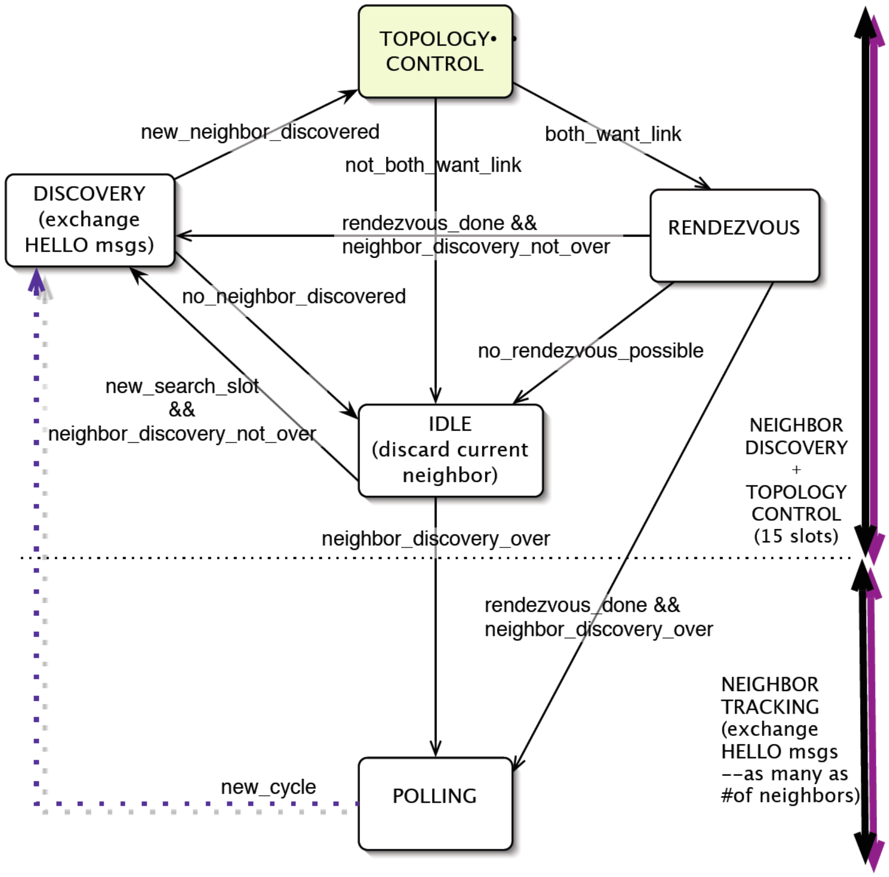


Figure 3.5: State diagram depicting the stages of our integrated scheme.

neighbor discovery in each cycle (Figure 3.5). Each pair of nodes that have a link will *rendezvous* at a common time and keep communicating on a periodic[7] basis. For this, we incorporate the polling phase proposed in [42]. If a node u does not repeatedly receive polling messages from w in the corresponding polling time, it assumes that the link is broken and removes w from its set of direct neighbors. This happens if w discovers a new neighbor z with a lower degree than u and abandons the link (w, u) to form a link with z.

[7] The frequency of these rendezvous instants depends on the speed with which the network topology changes.

Proposition 3.3.1 *The proposed scheme can also be implemented with beam-switching antennas; these antennas may be preferable as a less expensive alternative to fully adaptive arrays. In this case, as opposed to the* exact *direction that maximizes the received signal power, nodes will determine the* closest *(in terms of angular separation from the intended direction) sector to the desired direction.*

3.4 Performance Evaluation

We examine the average-case performances of our distributed algorithms D-LDS and Di-ATC[8] using simulations; we point out how our different design goals are reflected in their relative performances. We analyzed the worst-case performance of D-LDS in Section 3.2.2; the performance evaluation in this section shows how its average-case performance compares with the worst-case bounds.

Simulation Setup

Simulation Environment and Communication Model: We implemented D-LDS in a C++ simulator. Our simulator incorporates the conditions that were assumed for the design of D-LDS. In particular, (a) nodes have continuous access to information with regards to their two-hop neighborhood, (b) directional antenna footprints are ideal and no side lobes are formed.

We implemented Di-ATC in OPNET v.11 [2], in order to evaluate its performance under more realistic communication models. OPNET allows the design of antenna patterns having arbitrary gain and shape; in addition, one can modify the boresight of the created antenna with specific system calls and point the main lobe at an

[8] We simulate Di-ATC, integrated with the fully directional neighbor discovery and the neighbor maintenance mechanisms.

arbitrary point on the plane; the side lobes are created automatically, given the beamwidth of the main lobe and the maximum gain in the pointed direction. Nodes sense the channel and transmit if the channel is idle. Upon each transmission in the pointed direction, the nodes within the directional footprint (that have their antennas pointed towards the transmitter) that receive the packet compute the signal-to-noise ratio (SNR) by considering the antenna gains and their positions relative to the transmitter. The corresponding bit error rate (BER) is computed based on this SNR value and the modulation scheme in use (we use BPSK in all scenarios); if the number of bit errors in the packet exceeds the error threshold that the decoder can accommodate, the packet is discarded at the MAC layer. This setup offers a fairly realistic communication model using directional antennas.

Input Topology: We consider randomly generated topologies of varying densities in the area of deployment. When we simulate Di-ATC for a specific antenna beamwidth, the "unit" represents the communication range for that beamwidth. D-LDS is evaluated on static scenarios for various node densities and various network sizes. Di-ATC is evaluated at different densities, but due to the limitations in the simulation software we do not simulate Di-ATC on very large networks. We also examine the performance of Di-ATC with node mobility (where nodes move according to the random waypoint mobility model).

Antenna-Specific Parameters for Di-ATC: All nodes use the same antenna model in a single simulation experiment; they all transmit with a fixed power in all experiments. The maximum gain in the main lobe is 20dB (it is fixed in all simulations); the communication range varies with beamwidth. The directional range, D, is roughly a factor of $(360/\beta)^{1/\alpha}$ times larger than the omni-directional range, where β is the antenna beamwidth and α is the path loss exponent; this model complies with that described in [75]. For the neighbor discovery phase we use the method suggested in [42]. In particular, we reserve 15 time slots for neighbor discovery in every cycle. The cycles repeat continuously, and each simulation is run for 80 cycles.

With Di-ATC, nodes maintain logical connectivity with all discovered neighbors until the degree bound is reached (unlike the behavior with D-LDS where a node would not keep two neighbors that have a low angular separation). Therefore, in the topologies formed by Di-ATC the average degree is close to the degree bound; this behavior is common in all scenarios. Because of this, in most of our experiments we report results with a fixed beamwidth of 45°. We also impose a degree bound of 6 unless specified otherwise, this value is motivated from the construction by D-LDS.

Parameters and Metrics

We test the dependance of our topology control algorithms on the following parameters.

1. *Node density.* We simulate both algorithms for various node densities in the region of interest. (Note that topology control is useful only in network deployments with moderate or high-densities.)

2. *Area of Deployment.* We vary the size of deployment area.

3. *Antenna Beamwidth.* We simulate Di-ATC using different antenna beamwidths (Due to the assumption of circular transmissions that sweep the unit disk, the performance of D-LDS is independent of the antenna beamwidth).

4. *Speed of Nodes.* We simulate Di-ATC in scenarios of low and high mobility.

5. *Degree Bound.* Di-ATC imposes the degree bound a priori; we observe how our metrics specified below change when the bound is changed.

We quantify the performance in terms of the following metrics:

1. *Average node degree.* The average of all nodes' degrees in the network.

2. *Node degree distribution.* We measure the percentage of nodes in the network that have a certain degree.

3. *Average hop stretch.* For every node pair u,v that could communicate in the initial topology, we measure the hop count of the shortest path between them in the constructed topology. We define this value to be the hop stretch for the particular edge. An average is simply computed over all node pairs that can communicate in the scenario of interest. With Di-ATC the hop stretch can be influenced by neighbor discovery; thus, we measure this metric relative to an *ideal* topology. "Ideal" topology implies that nodes can discover *all* nodes with which they can communicate fully-directionally, *and* maintain *all* discovered neighbors. (This topology would include the shortest paths between all node pairs.)

4. *Maximum 95% hop stretch.* We quantify the 95% tail[9] of the hop stretch of all edges in the considered input graph.

5. *Hop stretch distribution.* The percentage of nodes having a certain hop stretch.

6. *Number of broadcasts.* We quantify the number of broadcasts performed during the execution of D-LDS. (As the MIS construction and acquiring the two-hop neighborhood are well studied problems [63],[19], we do not include their broadcast overhead in quantifying this metric.)

[9] x% tail of a set of values, is the value that is bigger than x% of the values in that set.

We believe that these metrics capture the effectiveness of D-LDS and Di-ATC quantifying the *trade-offs* between node degree and hop stretch.

Results and Discussion

In the rest of this section:

- We examine and compare the topologies formed by D-LDS and Di-ATC, in terms of *node degree* and *hop stretch*.

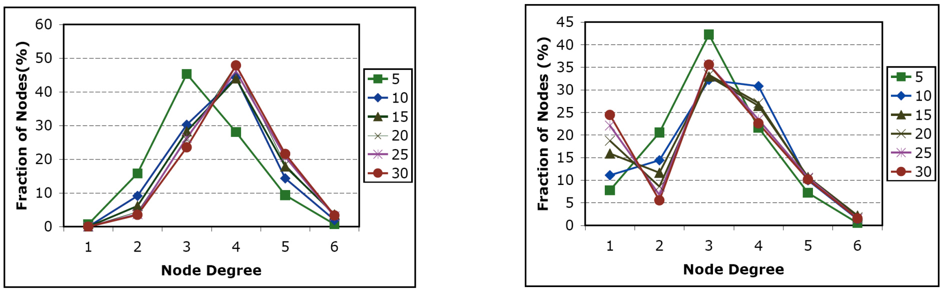
- We study the impact of changing the directional antenna beamwidth on the performance (in terms of node degree and hop stretch) of Di-ATC. (Since D-LDS assumes that nodes perform circular transmissions to sweep their neighborhood, its performance is independent of antenna beamwidth.)

- We study the node degree and hop stretch with Di-ATC in mobile scenarios.

- We quantify the message complexity of D-LDS with (a) reliable *and* (b) lossy links. With D-LDS, nodes' topology control decisions rely on accurate two-hop neighborhood knowledge; our objective here is to compute the cost of acquiring this information. Note that this is not required for Di-ATC operations.

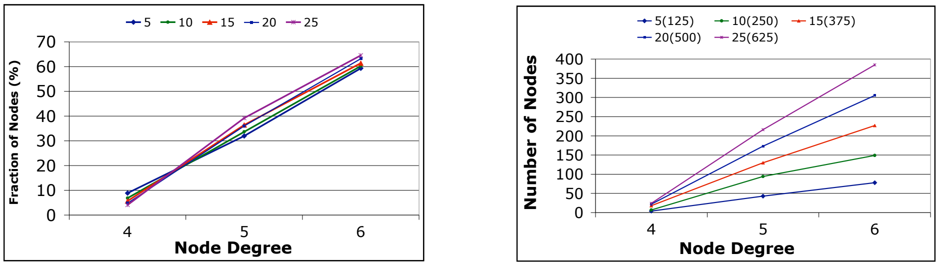
- We study the impact of link losses (and the impact of resultant delay in making topology control decisions) on the connectivity of the topology formed by D-LDS.

Node Degree Performance: We know from Theorem 3.2.25 that the nodes in the topology constructed by D-LDS have a *maximum* degree of 6; we simulate D-LDS and observe the distribution of node degrees in the topology formed. Figure 3.6 shows the degree distribution in the backbone and in the overall topology given input graphs



(a) Distribution on the Backbone (b) Distribution in Overall Topology

Figure 3.6: Node degree distribution in the topology formed by D-LDS.



(a) Fraction of Nodes (b) Number of Nodes

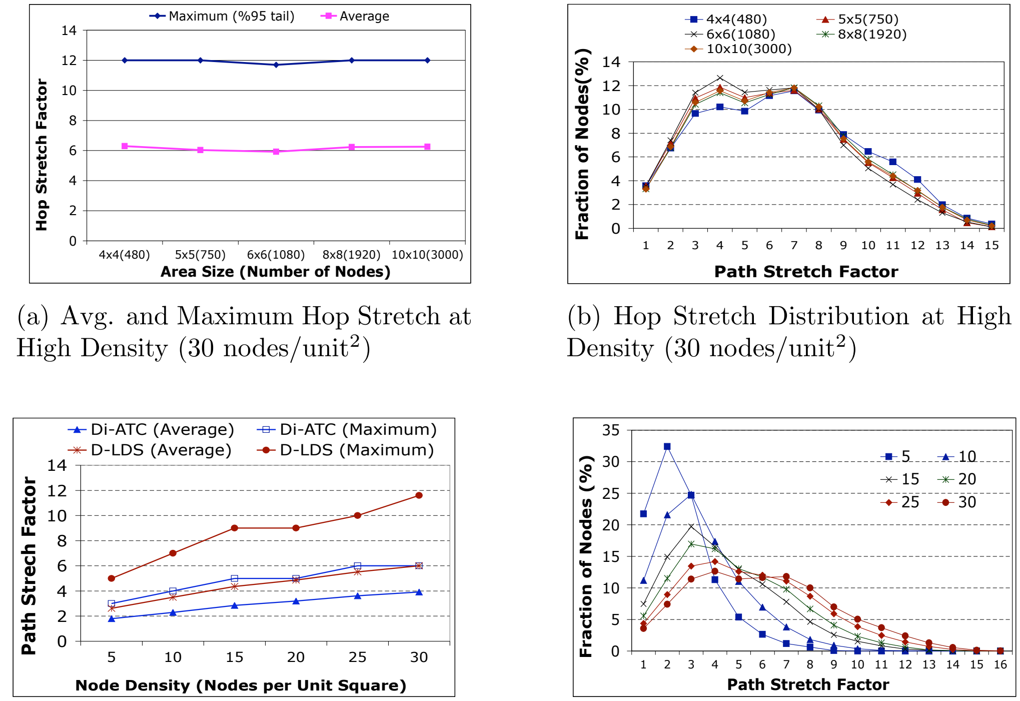
Figure 3.7: Node degree distribution in the topology formed by Di-ATC.

of various node densities (from 5 to 30 nodes/unit^2 on the 6x6 region). As seen in Figure 3.6(a), almost half of the backbone nodes have degrees of at most 4 at node densities of 10 and above. (The UDG is quite sparse at a node density of 5; hence nodes have lower degrees as compared to the other densities.) It is also important to note that not more than 4% of the nodes on the backbone have a degree equal to the worst case bound of 6. As the balanced binary tree nodes have well defined degrees (half of them have degree 3 and the other half, i.e. the leaves, have degrees of 1), the distribution of nodes in the final topology is biased towards lower degrees in Figure 3.6(b).

We also examine the degree distribution in the topologies generated by Di-ATC for input networks with different densities (nodes are deployed in a 5x5 unit area due to limitations in the ability of OPNET in performing large scale simulations).

We plot the percentage of nodes having a particular degree in Figure 3.7(a), and the actual number of nodes having that degree in Figure 3.7(b). With Di-ATC most nodes have degrees equal to the maximum of 6 and overall, the node degrees are higher in comparison to the degrees with D-LDS. This is because Di-ATC allows a node u to keep a discovered neighbor z, (irrespective of its angular separation with its existing neighbors) as long as the degree bound is not exceeded. On the other hand, the average degree with D-LDS is lower because *(i)* backbone nodes are not allowed to keep two neighbors that violate the angular separation constraint, and *(ii)* the nodes along the trees have a maximum degree of 4 (degree 4 occurring only at the root). With D-LDS, the average degree is slightly lower at higher densities, as the percentage of tree nodes increases. Overall, node density has little effect on the performance with both schemes. (Note here that, in these experiments with Di-ATC, we run the simulations long enough to facilitate discovery of the directional neighborhood; thus, the network converges and the effect of neighbor discovery on the topology control performance is reduced).

Hop Stretch Performance: In addition to ensuring low node degrees, both our schemes yield **low hop stretch**. The topology constructed by D-LDS, for all considered networks of high density (as we plot in Figure 3.8(a)), is a hop-spanner with a worst case maximum 95 percentile hop stretch of 12. In the same figure we also show that the average hop stretch is close to 6 for *all* network sizes that were considered. These results demonstrate that the performance of D-LDS *scales* in moderately sized to large networks. We also measure the average and the 95 percentile hop stretch as a function of node density. (the area of deployment is fixed at 6x6 units). We observe that the hop stretch is even lower at the moderate node densities that are more likely in ad hoc network deployments (Figure 3.8(c)). In Figure 3.8(b)



(a) Avg. and Maximum Hop Stretch at High Density (30 nodes/unit^2)

(b) Hop Stretch Distribution at High Density (30 nodes/unit^2)

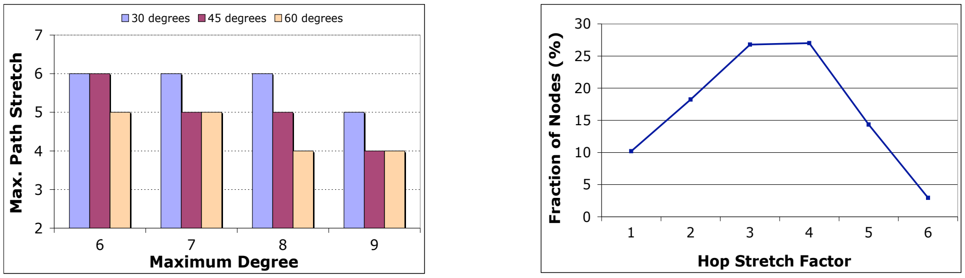
(c) Avg. and Maximum Hop Stretch at Various Densities

(d) Hop Stretch Distribution at Various Densities

Figure 3.8: Hop stretch for the topologies generated by D-LDS and Di-ATC.

and 3.8(d) we show the the percentage of node pairs that have a specific hop stretch in a given network.

In Figure 3.8(c) we also present the average and maximum hop stretch with Di-ATC given networks with different node densities; in this figure one can visualize how these results compare with the results from D-LDS. Both the maximum and the average hop stretch with Di-ATC are lower compared to the corresponding results with D-LDS. This is due to the higher average degree in the topology formed by Di-ATC (Figures 3.6 and 3.7). Hop stretch increases in denser deployments, as the output topology includes a smaller ratio of the total number of links that are in the ideal topology. This increase is less visible with Di-ATC, as nodes maintain a larger subset of their discovered neighbors than they do when they execute D-LDS.



(a) Maximum Hop Stretch

(b) Hop Stretch Distribution (45°-beamwidth)

Figure 3.9: Hop stretch performance of Di-ATC.

Next, we examine the impact of angular separation on the hop stretch with Di-ATC by varying the imposed degree bound. We experiment with different antenna beamwidths; the hop stretch with each beamwidth is computed relative to the corresponding *ideal* topology (discussed earlier). We present the maximum 95% and average values of hop stretch in Figure 3.9. As expected, the hop stretch is less severe when the degree bound requirement is less strict (Figure 3.9(a)).

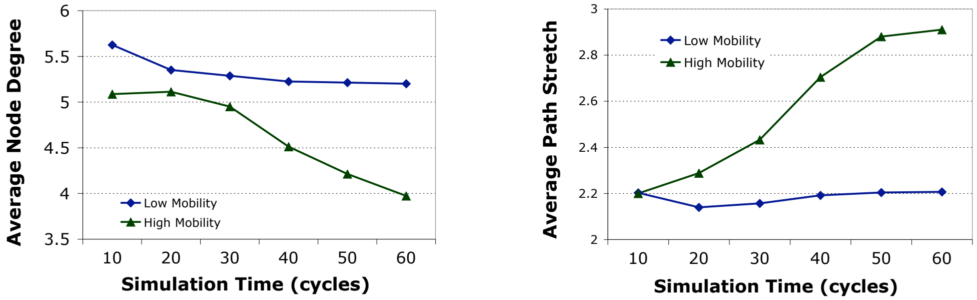
Figure 3.9(a) shows that for a given degree bound the hop stretch is higher when narrower directional beams are used. Probability of neighbor discovery increases with a larger beam; nodes have a greater degree of freedom in choosing the angularly separated subset of their discovered neighbors. However, note that due to the larger radial range with a narrower-beam directional antenna, a node can potentially reach (though it may not discover) more neighbors with a narrower beamwidth; thus, the corresponding ideal topology includes more links than it would with a larger beamwidth. As a consequence, the hop stretch in the output topology G' is higher, as more links that make up the shorter paths in the ideal topology do not exist in G'.

Figure 3.9(b) shows that more than half of the paths in the topology formed by Di-ATC have a stretch of less than or equal to 4; this result can be ascribed to the relatively high node degrees resulting with Di-ATC unlike with D-LDS (Figures

3.6,3.7).

Performance in Mobile Scenarios:

We simulate Di-ATC in two mobile scenarios with the random waypoint mobility model. Nodes choose a random speed between (0,10] m/sec; this represents a scenario in which the topology changes are slow. In a second scenario nodes choose speeds in (10,20] m/sec for a faster-changing topology. In both scenarios nodes pause for 1 sec at each intermediate destination point. The 1 sec. pause time corresponds to a duration of 2.5 cycles (Recall our discussion of cycles in Section 3.3). As nodes' coordinates are dynamic due to mobility, we compute the hop stretch for the mobile case in a different fashion. At particular time instants, we find shortest path between two nodes (u,v) in the snapshot of the constructed topology at that time instance, and compare it with the shortest path (between u,v) in the ideal topology consisting of the links that could potentially exist at that time (as before, the ideal case assumes that the complete neighborhood is discovered and all links are maintained).



(a) Average Degree vs. Time with Node Mobility)

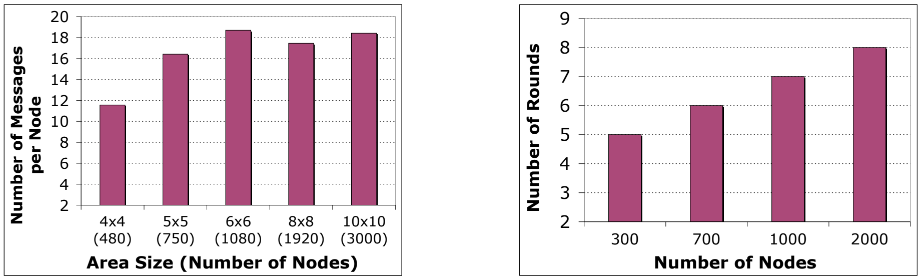
(b) Avg. Hop Stretch vs. Time with Node Mobility)

Figure 3.10: Average node degree and hop stretch in the topologies generated by Di-ATC in existence of low and high node mobility.

Figure 3.10 shows the average degree and hop stretch in a mobile network, at arbitrarily selected cycle instants in time during a simulation run. Figure 3.10(a) shows that the average degree does not fluctuate much after stabilizing at cycle 40 when mobility is low; with higher node mobility the average degree drops after the first half of the simulation, as nodes lose their neighbors due to high relative speeds (it becomes very hard to track neighbors). Note however that, these high speeds are not common in typical network deployments. In fact, at these extremely high speeds, use of steerable directional antennas with narrow beams is challenging.

We observe that the hop stretch also stabilizes around cycle 40 (Figure 3.10(b)). The initial period until stabilization is spent for discovering neighbors. The hop stretch exhibits a slight increase after cycle 30 in the case of fast mobility, due to the decrease in node degrees. The low hop stretch can be attributed to the angular separation among neighbors of a node, which helps Di-ATC construct sparse but well-spread topologies.

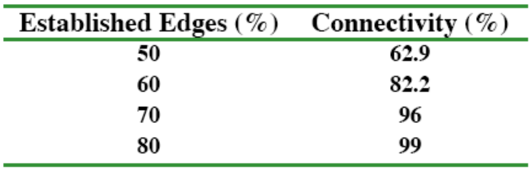
Communication and Time Complexity of D-LDS: We showed in Section 3.2.2 that the number of messages required by D-LDS is $O(n)$ i.e., $= cn$, where c is a constant. The simulations, however, can provide an estimate of the hidden constant c. To quantify c, we generate enough nodes such that there are approximately 30 nodes in every square unit; this is considered a dense ad hoc network. Fixing this density, the input network is varied from 480 nodes in a 4x4 unit area (a moderate-size ad hoc network), to 3000 nodes in a 10x10 unit area (a large ad hoc network). We measure



(a) Message Complexity of D-LDS

(b) Time Convergence of D-LDS

Figure 3.11: Message and time complexity with D-LDS in various topologies.



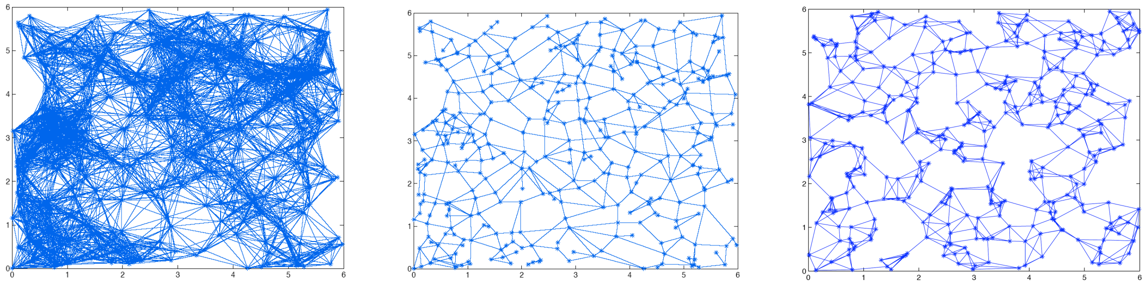
Established Edges (%)	Connectivity (%)
50	62.9
60	82.2
70	96
80	99

Table 3.1: Connectivity at different stages of the D-LDS execution.

the total number of broadcasts used by D-LDS in constructing the final topology, and divide it by the number of nodes. The results are depicted in Figure 3.11(a); they show that the measured estimate of the constant stabilizes to between 18 and 19 and does not change even when the number of nodes is as high as 3000.

We also examine the time it takes until the backbone converges to a 9-spanner. This provides us with an idea of the parallelism possible with D-LDS. We define convergence time in terms of "the number of rounds", where a round corresponds to the duration of transmission of a single message. Since with D-LDS, edges can be inserted in parallel, we expect the time taken for convergence to be much smaller than the worst case $O(n)$ bound. To corroborate this expectation, we construct topologies of varying densities in a 6x6 unit area. We depict the results in Figure 3.11(b). We observe that even with a large number of nodes (as high as 2000), convergence takes 8 rounds. This demonstrates the high degree of parallelism that is possible with D-LDS.

Connectivity with D-LDS with Lossy Links: The packet losses at the link layer can delay the establishment of edges with D-LDS. We investigate how the connectivity of the network is affected by this delay. In other words, we study the convergence of D-LDS in terms of forming the final graph. To understand this, we perform the following experiment. 1000 nodes are placed in a 6x6 unit area and trace the connectivity of the backbone at different stages of D-LDS execution; at each stage, note that only a percentage of the final set of edges is established. We investigate the impact on the backbone connectivity only; the binary balanced trees are constructed



(a) Input Topology

(b) Topology Generated by D-LDS

(c) Topology Generated by Di-ATC

Figure 3.12: Visualization of the input topology as a Unit Disk Graph (3.12(a)) and the topologies generated by D-LDS (3.12(b)) and Di-ATC (3.12(c)). (UDG consists of 360 nodes in a 6 by 6 area.)

by nodes that are within the one hop reach of each other, and hence, do not affect the connectivity of the other nodes beyond their group.

The results of the experiment are shown in Table 3.1. Note that D-LDS provides connectivity even when just a subset of edges are established; connectivity is guaranteed even if only 70% of the edges are established.

Summary: we have evaluated the performance of our distributed topology control algorithms D-LDS and Di-ATC, in various scenarios. We have showed their performances scale in terms of node density, network size, and area of deployment. Figure 3.12 illustrates a randomly[10] generated topology, and the topologies constructed by D-LDS and Di-ATC. The initial graph consists of 360 nodes in a 6x6 unit area. The unit with D-LDS is as defined in the beginning of this section; with Di-ATC we use 45°-beamwidth directional antennas and map its directional range to a unit distance.

[10] While our simulations function on input topologies where we generate nodes randomly using a uniform distribution, the bounds that were derived previously hold for **arbitrary** unit disk graphs.

3.5 Conclusions

Nodes in a fully directional ad hoc network need to continuously track their neighbors for maintenance. In this chapter, we proposed topology control strategies in order to reduce the overhead incurred due to this process.

We first designed the LDS algorithm that finds a sparse subgraph of a given arbitrary unit disk graph G, such that the maximum degree is **6** and the maximum hop stretch is $O(1 + \log \Delta)$, where Δ is the maximum degree in G. We show that this result is **near-optimal**.

Next, we designed the distributed, localized algorithm D-LDS, which forms a sparse topology having the same attractive properties as LDS using a linear number of broadcasts per node.

Finally, we relaxed the idealized assumptions we made to facilitate the design of LDS and D-LDS, and we designed a more practical, considerably simpler topology control scheme Di-ATC. Di-ATC integrates fully-directional neighbor discovery with the topology control decisions, and neighbor maintenance mechanism is applied only to a selected subset of discovered neighbors. We simulated the integrated framework in practical settings and studied the average performance to understand the tradeoff between node degree and hop stretch.

In the next chapter we will study the Multiple-Input Multiple-Output (MIMO) communication paradigm that is formed with antenna arrays on the nodes.

Chapter 4

The Modeling of MIMO Diversity Gain

Space-Time Block Coding (STBC) in a MIMO system provides significant robustness to the communication among a node pair. In this chapter, we shed light on the cross-layer interactions between the physical, link and routing layers in wireless multi-hop networks using MIMO-STBC communications. As we have stated in Chapter 2, many previous studies assume an overly simplistic physical layer model of MIMO-STBC that does not sufficiently capture these cross-layer interactions. We first observe these interactions in practice in Section 4.2. Next, we contrast the measurement-based observations with a simple model that has been commonly used by prior studies. We identify the physical layer features that are not considered in this model; we progressively incorporate these features to create new models, and compare each new model with the measurements. Four models are presented in this chapter.

4.1 Introduction

The use of STBC with antenna arrays offers a significant improvement in data transmission reliability in comparison to the traditional single antenna or SISO (single-input single-output) communications. This benefit on MIMO-STBC links is referred to as the *MIMO diversity gain*. Although diversity gain motivates the design of higher layer protocols to facilitate end-to-end benefits from the MIMO physical layer, to date, the diversity benefits due to MIMO have not been fully exported to the upper layers due to complex cross-layer interactions.

In order to understand the experimentally observed behavior of higher layer protocols built over MIMO, we take a slightly unconventional route. We first observe the behavior of MIMO-STBC links in practice, on our indoor 802.11n testbed. These measurements aid us to form a measurement-driven representation of MIMO-STBC in practice; this representation serves as a benchmark against which we assess the models. Next, we start our exploration by simulating a simple physical layer model of MIMO, and comparing it with the measurement-driven representation. We progressively build more complex models via bit-level simulations that incorporate additional physical layer features. Many of the key features that we add are missing in existing models [86, 41, 29].

4.2 Diversity Gain in Practice

In this section, we provide a brief introduction of the current IEEE 802.11n draft.

4.2.1 The IEEE 802.11n Draft

The IEEE 802.11n specification supports both STBC and space-division multiplexing (SDM) modes of MIMO communications [1]. 24 different data rates can be achieved with 802.11n using different numbers of spatial streams and with different modulation and coding schemes. 802.11n also supports the combination of adjacent channels (of 20 MHz each) to form a wider channel of 40 MHz; this technique is called channel bonding.

In our work, we focus on the STBC mode of operation. In order to compare SISO and MIMO performance on our testbed, we operate using the 20 MHz bandwidth in the 2.4 GHz band. We consider the FEC code rate R=3/4 with 4-QAM, 16-QAM and 64-QAM modulation schemes; the corresponding data rates with 802.11n are 19.5 Mbps, 39 Mbps, and 58.5 Mbps, respectively. With 802.11g, the corresponding data rates are 18, 36 and 54 Mbps, due to a different channel spacing. These map onto rates of 1 bps/Hz, 2 bps/Hz and 3 bps/Hz with the two specifications.

In this section, we describe our measurements on an 802.11n indoor experimental network. Our measurements serve as benchmarks for evaluating the fidelity of the MIMO-STBC simulation models that are considered in this work.

4.2.2 Our 802.11n Testbed and Measurement Setup

Our testbed includes 7 Dell S530 desktop PCs with 2.5 GHz Intel dual-core processor and 1 GB RAM. The deployment is depicted in Fig. 4.1.

Each node is equipped with an RT2860 miniPCI card that supports 802.11a/g/n, and an EMP-8602 Atheros-based card that supports 802.11a/g. We use 5 dBi gain onmidirectional antennae for these cards, and by default we use a transmission power of 20 dBm. We have modified the Linux client driver v1.6 [3] to enable 2x2 STBC

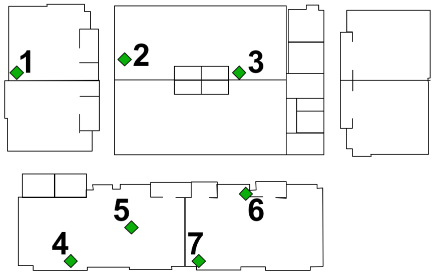


Figure 4.1: A snapshot of our 7-node 802.11n testbed. Nodes are represented by diamonds along with their IDs.

support. For this, we include the line:

```
{"HtStbc",    Set_HtStbc_Proc},
```

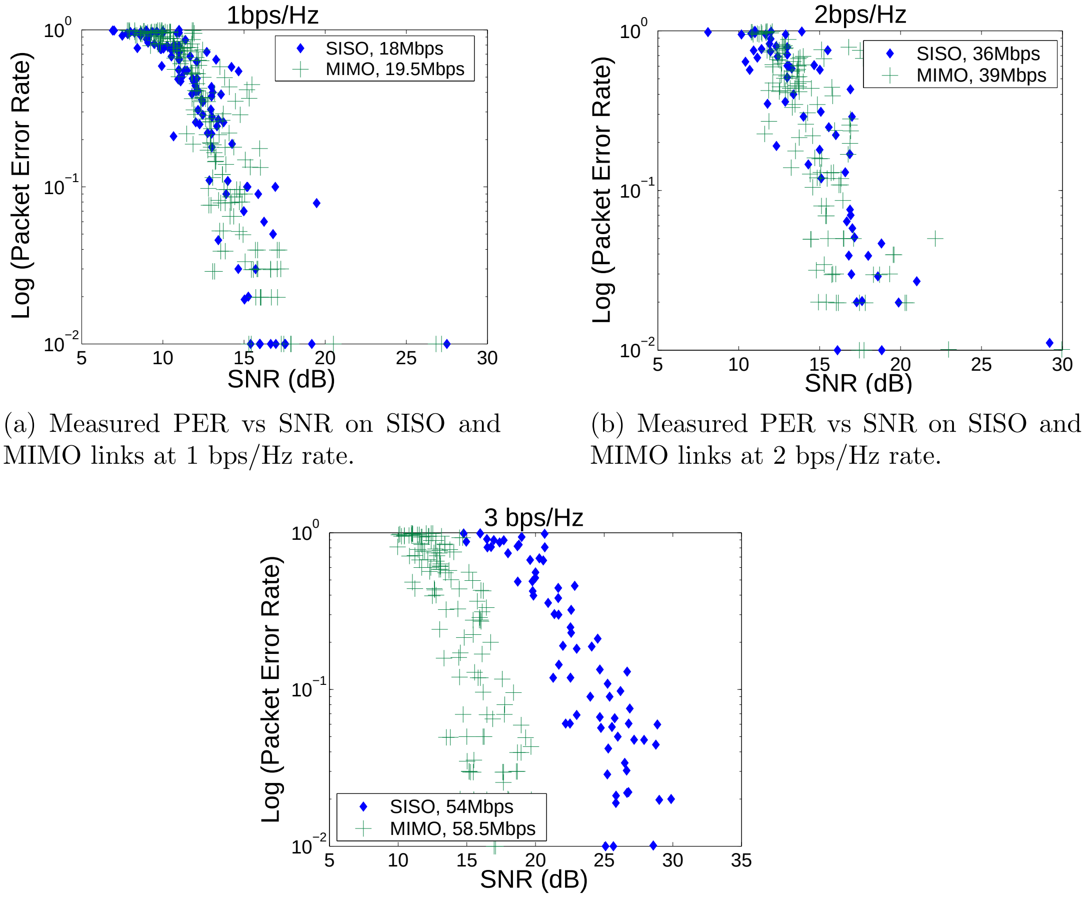
into the `RTMP_PRIVATE_SUPPORT_PROC` struct array, located in *os/linux/sta_ioctl.c*. We carry out link-level measurements and examine the measured data in terms of both bit-level and packet-level statistics.

We collect packet level statistics on both SISO and MIMO links. We relocate our MIMO nodes to create 33 different transceiver pairs (at different locations) and adjust their powers to get a rich set of channel properties and SNR values (we take measurements with different static topologies). The measurements consider both (a) line-of-sight (LoS) links where the communicating nodes are not separated by walls or other large obstacles, and (b) non-line-of-sight links (NLoS) where the end nodes are located in different rooms in our building. The same set of links is used in SISO and MIMO modes. We generate and transmit 10000 packets on every link; we measure the PER and RSSI (Received Signal Strength Indicator) based on the reportings of our cards[1]. The RSSI is an average of the signal strength measured at each antenna. We

[1] The cards operate at a constant noise floor level of -95 dBm, with an accuracy of 99.5%.

conduct our experiments late at night in order to avoid interference from co-located WLANs. For each link, we conduct the MIMO measurements right after the SISO measurements in order to reduce the possibility of significant variations in channel quality.

4.2.3 A Measurement-Driven Model of Diversity Gain (MDM)



(a) Measured PER vs SNR on SISO and MIMO links at 1 bps/Hz rate.

(b) Measured PER vs SNR on SISO and MIMO links at 2 bps/Hz rate.

(c) Measured PER vs SNR on SISO and MIMO links at 3 bps/Hz rate.

Figure 4.2: A comparison of the PER, measured at different SNR values on SISO and MIMO links using three different rates. The greatest gain with MIMO compared to SISO is observed at the highest rate.

Fig. 4.2 presents the measured PER for a range of SNR values at different rates, using SISO and MIMO modes. Notice that a significant performance improvement with MIMO over SISO is only observed at the higher rate of 3 bps/Hz (unlike what has been assumed in [86] and [41]). For a desired PER, the SNR requirement only decreases by about 1 dB at transmission rates of 1 bps/Hz and 2 bps/Hz; the decrease is much more significant at 3 bps/Hz. Fig. 4.3 depicts the averages of the PER values for each SNR value from our measured results (see Fig. 4.2). We deliberate on the reasons for this observation in Section 4.3.

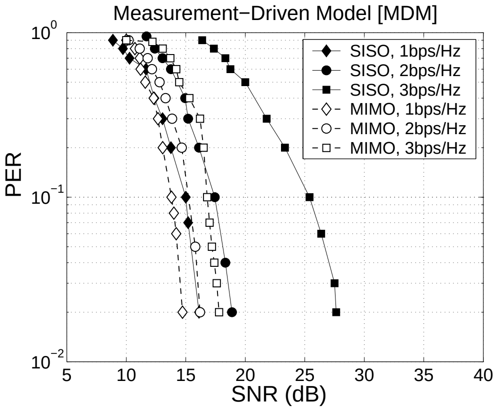
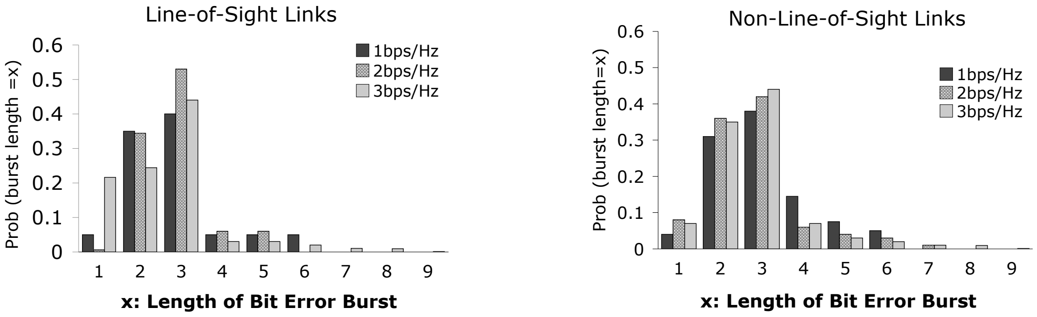


Figure 4.3: The Measurement-Driven Model (MDM). The curves are computed as averages over the points in Figures 4.2(a)-4.2(c); they demonstrate the average PER-SNR trend of MIMO-STBC links on the testbed.

Bit Error Statistics: We also examine the burstiness of errors in our indoor setting by conducting bit-level measurements at the MAC layer. In these experiments we utilize our *EMP-8602 6G* cards. We modify the MadWifi driver (version 0.9.2) [4] to disable the CRC (cyclic redundancy check) at the MAC layer, in order to measure the BER observed at the MAC layer. We transmit pseudo-random payloads; this allows us to identify the bit errors at the receiver. Our measurements suggest that bit errors are likely to occur in bursts of varying length on both LoS and NLoS links, as shown in Figure 4.4. The burstiness of errors depends on the coherence time of the channel i.e., the duration for which the channel quality does not change signficantly. Similar results have been reported in [64] for SISO links. Note that at the PHY layer (prior to decoding), the error patterns in coded bits may be different from that observed at the MAC layer; due to the presence of an interleaver/de-interleaver, one could expect both types of errors at the PHY layer. As we discuss in Section 4.3, the burstiness of bit errors is one of the contributing factors to the diminished gains with MIMO at the low rates.



(a) (b)

Figure 4.4: Distribution of error burst length on the LoS and NLoS links in our testbed, measured at different rates.

4.3 Seeking For an Accurate Model of Diversity Gain

In this section we describe four different models of the space-time coded MIMO physical layer; each model reflects the relation between PER and SNR. We begin with a simple model, variants of which have been previously used in [86],[41] and [29]. Each of the subsequent models studied in this section incorporates additional PHY layer components[2].

We mainly examine the models at three bit rates of 1, 2 and 3 bps/Hz. The choice of these rates enables us to observe the behavioral nuances of MIMO diversity. In particular, our observations in Section 4.2 suggest that at low rates MIMO does not offer significant benefits. With transmission rates that are lower than 18 Mbps in 802.11g (*i.e.,* 1 bps/Hz), we do not observe MIMO diversity gain. On the other hand, 54 Mbps (3 bps/Hz) is the highest possible transmission rate with SISO. We expect the gains due to MIMO diversity to be more pronounced at even higher rates; however, we use at most 3 bps/Hz transmission rate in order to be able to compare MIMO and SISO link behaviors to evaluate the impact of diversity gain. Finally, we pick 36 Mbps (2 bps/Hz), since it is the median rate between 18 and 54 Mbps.

In the following, each model represents the relation between the reception SNR and the average BER or PER at this rate, on SISO and MIMO links. The relative behavior of these two systems yields how the diversity gain changes with SNR, as captured by the representing model.

4.3.1 Model 1 (M-1): Constant Gain at a Given Rate

With M-1, a constant diversity gain is attributed to all (2x2) MIMO links that use Alamouti codes at a certain transmission rate. In particular, based on M-1, if the SNR at the receiver of a SISO link is Γ_{SISO} dB and the corresponding BER is α, the same SNR would achieve a BER of β (where $\beta < \alpha$) if the link were MIMO. Here β corresponds to the BER that would be achieved with an SNR of $\Gamma_{SISO} + d$ dB on the SISO link. d, the difference in the two SNR values for achieving a target BER of β (on the MIMO and SISO links) models the diversity gain by M-1. With M-1, d is a

[2] Whenever we move to a new model with an added PHY layer component, we also incorporate the new component in the evaluation of the corresponding SISO system.

fixed value and it is directly applied at the packet level. Hence, the MIMO link has a PER at an average SNR value, which is d dB smaller than the SNR that achieves the same PER on a SISO link.

Applying M-1: In [29], d is quantified based on the bit-level simulations reported in [39]. In [39], the Alamouti-coded 2x2 MIMO system is used with BPSK, QPSK and 8-PSK modulation schemes, respectively, in order to achieve the rates of 1, 2 and 3 bps/Hz. From the reported BER vs SNR trend curves, we retrieve the value of d for a target $\beta = 10^{-3}$. These values are shown in Table 4.1.

Data Rate	**SISO SNR at BER=10^{-3}**	**MIMO SNR at BER=10^{-3}**	**Diversity Gain**
1 bps/Hz	24 dB	7 dB	17 dB
2 bps/Hz	27.08 dB	14 dB	13.08 dB
3 bps/Hz	31.25 dB	18.75 dB	12.5 dB

Table 4.1: Diversity Gain with M-1

At the higher layers, diversity gain on a link is observed in terms of a reduction in the PER, owing to an increase of d dB in the SNR. The variants of this abstraction have been used in previous research efforts that examine the impact of diversity gain on the multi-hop network performance [29],[41],[86].

4.3.2 Model 2 (M-2): Diversity Gain Depends on the Reception SNR

M-1 assumes that the achievable diversity gain, d, is fixed at a given rate. However, our observations on the testbed (Section 4.2) suggest that the value of d may vary for different Γ_{SISO} values. In particular, the advantage due to MIMO (at the physical layer) is more pronounced in the higher SNR regimes, and the diversity gain is much lower when the reception SNR is low. In addition, the BER achieved at a given SNR increases if the link operates at a higher data rate.

The model M-2 captures the above-mentioned characteristics of the diversity gain. In modeling M-2, the SNR-BER trend in [39][3] is quantized into bins of SNR intervals of 1 dB. At a given rate, for a reception SNR of Γ_{SISO} dB, the bin that includes this value provides the achievable BER at the corresponding Γ_{MIMO} dB. Thus, as opposed to a constant *increase* of the SNR (as with M-1), the achievable diversity gain is quantified as a *reduction* of BER. The gap between the Γ_{MIMO} and Γ_{SISO} that achieve the same BER, β, is not fixed; it depends on β. For illustration, we tabulate a sub-set of the quantized levels in Table 4.2; the table represents the diversity gain at SNR values between 8dB and 17dB, at three different rates. The quantized BER-SNR trend at three different rates is plotted as continuous curves in Fig. 4.5(a). We observe from the table and the figure that as the SNR increases, the difference between the achieved BER with MIMO relative to SISO increases. Furthermore, for a given SNR, the reduction in the BER is more significant at the lower rates.

Applying M-2: In order to carry over the diversity gain to the higher layers we first map the SNR to a BER. Subsequently, when the packet is received with a certain SNR, random bit errors are placed within the received frame according to a Bernoulli distribution with a probability equal to the BER. The packet is deemed successful only if the total number of bit errors is zero.

Transition to higher fidelity models: The curves in Figure 4.5(a) suggest that at the low rate of 1 bps/Hz, the MIMO system achieves a target BER of 10^{-3} with an SNR value that is ∼20 dB less than the SNR of the SISO system that achieves the same BER. However, measurements from Section 4.2 do not quantify such high gains

[3] Section 4.10, page 101 in [39].

	1 bps/Hz		**2 bps/Hz**		**3 bps/Hz**	
Reception SNR	**MIMO BER**	**SISO BER**	**MIMO BER**	**SISO BER**	**MIMO BER**	**SISO BER**
8-9 dB	0.0002.5	0.03	0.004	0.05	0.02	0.085
9-10 dB	0.0001	0.022	0.001	0.04	0.014	0.088
13-14 dB	5.0E-6	0.009	5.0E-5	0.02	0.0017	0.04
14-15 dB	2.1E-6	0.008	2.2E-5	0.018	0.0009	0.03
16-17 dB	5.0E-7	0.005	4.0E-6	0.0095	0.0002	0.021

Table 4.2: Diversity Gain with M-2

(Figure 4.3). Although M-2 is more accurate than M-1 in depicting the behavior of MIMO diversity, it represents a PHY layer that is different from that described in the 802.11n draft [1] in two main ways:

1. The modulation schemes used in 802.11n to generate rates of 1, 2 and 3 bps/Hz are 4-QAM, 16-QAM and 64-QAM (and not BPSK, QPSK and 8-PSK as assumed in [39]).

2. The measured results correspond to a system where convolutional FEC codes are employed; the results in [39] do not assume any FEC codes.

As a result, we improve the model M-2 by incorporating the PHY layer components of the practical system (*i.e.,* 802.11n communications) and therefore build a model that has a *higher fidelity* with respect to the 802.11n system. To obtain these models we perform bit-level simulations of the system whose parameters are specified in Table 4.3.

4.3.3 Bit-Level Simulator for Generating Higher-Fidelity Models

Bit-level simulations are performed on MATLAB. We implement the following modules, whose parameters are specified in Table 4.3: random bit generation, convolutional encoding, bit-to-QAM symbol mapping, space-time block coding (STBC), slow and fast Rayleigh fading generation, noise generation, STBC decoding, log-likelihood ratio (LLR) computation, and Viterbi decoding.

Random information bits (0 or 1) are generated using the *randint()* function in Matlab. These information bits are organized into packets of size $N_b = 1500$ bytes (to conform with 802.11n). The information bits are then convolutionally encoded using a rate-1/2, constraint length 7 convolutional code. The rate-3/4 code is obtained by puncturing the rate-1/2 code. The convolutional encoding is implemented using the *convenc()* function in Matlab. With reference to Table 4.3, the constraint length of a code refers to the number of input (6 past and 1 present) data values that are used to generate the code; Generator polynomials are used to construct the coded output symbols. As per the IEEE standards, the generator polynomials are 171 and 133 (in octal). The output bits from the convolutional encoder are mapped to M-QAM-symbols. 4-QAM, 16-QAM and 64-QAM symbols with Gray mapping are used in the simulations. These QAM symbols are then fed to the STBC encoder. We have implemented the 2×2 Alamouti code (2-Tx antennas, 2-time slots) for space-time encoding. The output of the STBC encoder is multiplied with Rayleigh (complex Gaussian) fading samples, following which AWGN (Additive White Gaussian Noise) samples are added. We observe these MIMO links on two extreme[4] channel conditions: a fast fading and a slow fading channel. To simulate a fast fading channel, we assume that the fade remains constant over one STBC block and varies independently from one STBC block to the other; this is widely referred to as the quasi-static assumption in the literature [39]. In the case of a slow fading channel, we assume that the fade remains constant over the entire packet (we refer to the subsequent 1500 bytes as a

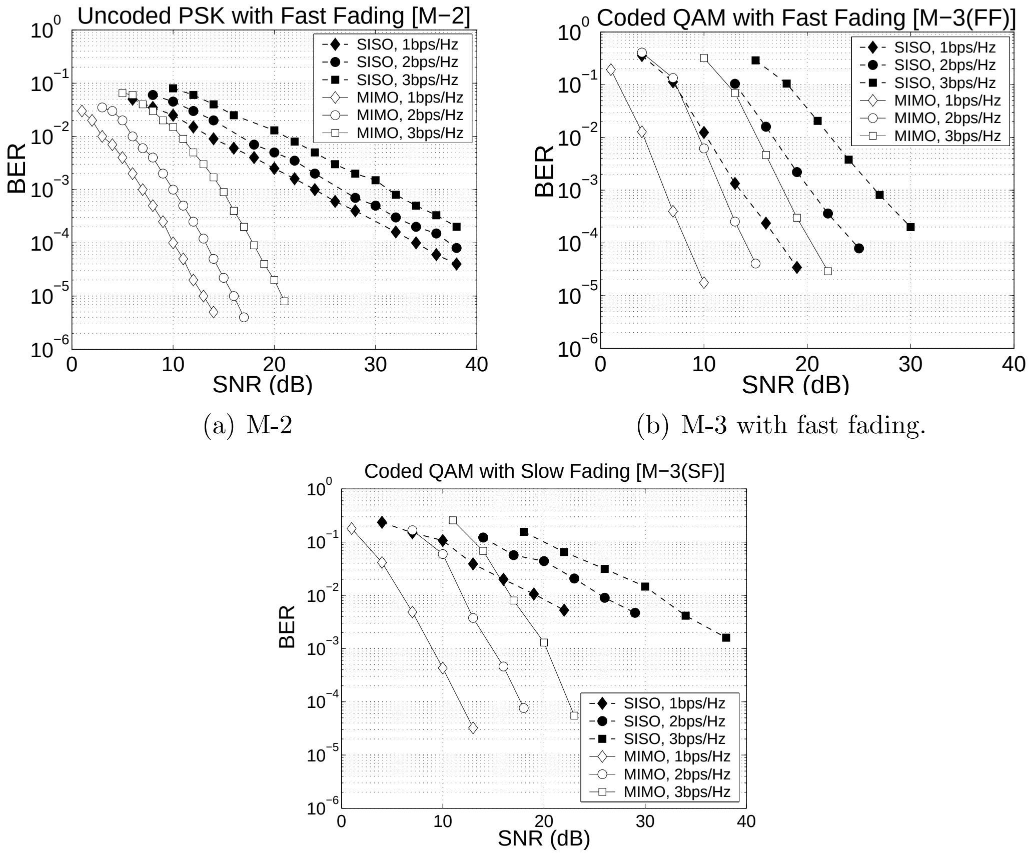
[4] In practice, the bit error distribution indicates fade durations that are between these two extreme cases, see Figure 4.4.

“packet”; this is the size of the packets used in the measurements in Section 4.2), and varies independently from one packet to the other. The SNR values in the simulations are varied by fixing the signal power and varying the noise variance. At the receiver side, we have implemented the maximum-likelihood (ML) decoder for STBC decoding (i.e., decoding of the Alamouti code). From the output of the STBC decoder, we compute the LLR values of the bits that form the QAM-symbols. These LLR values are then fed as soft inputs to the Viterbi decoder for decoding the convolutional code. The *vitdec()* function in Matlab is used for the Viterbi decoding. We calculate the bit error rate (BER) and the packet error rate (PER) at the output of the Viterbi decoder. In evaluating the PER, we consider a packet to be in error even if one bit in the packet is in error; i.e., a packet is considered to be successfully received only if all the bits in the packet are received correctly.

4.3.4 Model 3 (M-3): Higher Consistency with 802.11n

Using the simulator described above, we create M-3, a new model that incorporates the modulation and coding schemes that are used in 802.11n (Table 4.3). We simulate M-3 with both fast fading and slow fading channels; we refer to M-3 with these channels as M-3 (FF) and M-3 (SF), respectively. In the former channel model we assume the fading characteristics are independent per bit. With the latter channel model, it is assumed that the duration of fades is greater than a symbol period; hence, the transmitted data symbols experience longer (i.e., time-correlated) fades. These are two extreme cases, and in reality the channel exhibits fade durations in between these values.

Applying M-3: The SNR-BER trend curves representing the (SISO and MIMO) link behaviors of M-3 (FF) and M-3 (SF) appear in Fig. 4.5(b) and Fig. 4.5(c),



(a) M-2

(b) M-3 with fast fading.

(c) M-3 with slow fading.

Figure 4.5: Bit-Level Models (M-3).

respectively. As with M-2, we quantize these BER-SNR results into bins of 1 dB each and form tables based on these figures for M-3 (FF) and M-3 (SF).

4.3.5 Packet-Level Representation of Model 3 (M-3P)

Despite these additional levels of sophistication in M-3, the SNR-BER trend curves in Figures 4.5(b) and 4.5(c) have a more gradual fall off with SNR compared to what is seen in the PER vs SNR curves that represent MDM (Figure 4.3). We conjecture that this is because the measurements provide PER as opposed to BER statistics.

System Component	**Specification**
Error Correction Coding	Convolutional Encoding Viterbi Decoding

Generator Polynomials of the Encoder	G0=$(171)_8$,
	G1=$(133)_8$

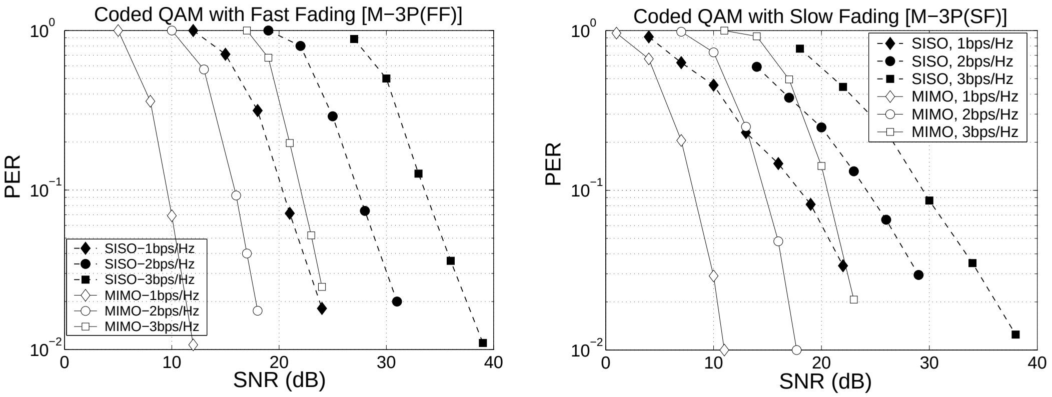
Encoder Constraint Length	7
Modulation schemes	QPSK, 16-QAM, 64-QAM
Code Rate	3/4 (with puncturing) for all modulations
Packet Length (for M-3P)	1500 bytes

Table 4.3: Specifics of our Bit-Level Simulator

As Figure 4.4 suggests, packet-level statistics may not directly follow from bit-level statistics, due to the bursty behavior of bit errors. Undoubtedly, for a given BER, *clustered* occurrence of bit errors lead to a *smaller* PER; or inversely, evenly spread out bit errors corrupt more packets.

In order to have a more meaningful comparison with the measured behavior on 802.11n MIMO links, we generate PER statistics using MATLAB simulations, as described earlier. We generate 1500-byte packets (as in our measurements), and discard a packet if any of its bits is in error. The obtained SNR-PER trend at different data rates constitute our packet-level model, M-3P. As in the case of bit-level models M-3, we derive two variants of M-3P called M-3P (FF) and M-3P (SF) corresponding to M-3P on the fast fading and slow fading channels, respectively.

Applying M-3P: The SNR-PER trend curves of M-3P (FF) and M-3P (SF) are shown respectively in Figures 4.6(a) and 4.6(b) for SISO and MIMO links at three different rates. As these figures provide packet-level statistics, using M-3P with the higher-layer simulations is straightforward; at a given rate, we discard a packet that is received with an SNR of Γ dB, with a probability that is equal to the PER at Γ dB for the given rate.



(a) PER vs SNR with a fast fading channel; these trend curves constitute the model M-3P (FF).

(b) PER vs SNR with a slow fading channel; these trend curves constitute the model M-3P (SF).

Figure 4.6: Packet-Level Models (M-3P).

4.4 Link-Level Comparison of Models and Measurements

In this section, we first discuss our observations from Figures 4.5 and 4.6. Next, we compare the link-level behavior represented by the four models described so far with that represented by the measurements from Section 4.2 (*i.e.,* the measurement-driven model, MDM).

From Figures 4.5, 4.6, and 4.3 we observe that the results with M-3P (SF) are more similar to MDM than with any of the other models. We make the following observations with regards to the higher-fidelity models presented in the previous section, namely the M-3 and M-3P.

- ***At a given SNR, the BER performance on SISO links with a slow fading channel is poorer compared to that with a fast fading channel:*** With fast fading, FEC codes are able to easily correct the independent occurrence of the bit errors. However, there is a high degree of correlation among the bits in error, on a slow fading channel; the bit errors occur in bursts. The duration of a fade, which reflects the size of the burst, is equal to the transmission duration of a packet. In this setting, FEC codes are not as successful as in the fast fading scenario. As a result, the SNR requirement for a target BER is higher with A-3 (SF) than with A-3 (FF) at all rates.

- ***At a given SNR, the BER performance on MIMO links with a slow fading channel is also poorer compared to the fast fading channel, but to a lesser extent:*** The performance with slow fading is worse than with fast fading, again due to the reduced effectiveness of FEC codes when errors are correlated. However, space-time diversity reduces the possibility of encountering deep fades. Consequently, fewer bits are in error and the degradation is less significant.

- ***On MIMO links, at a given SNR, the PER with M-3P (SF) is smaller compared to the PER with M-3P (FF):*** *This is a surprising result since the BER was higher with M-3 (SF) compared to M-3 (FF).* This result is an artifact of two effects. On the one hand, the FEC codes are less effective with a slow fading channel than with a fast fading channel. However, the errors are confined to fewer packets (than with fast fading where the errors are more spread out). The first effect causes an increase in the PER, while the second effect reduces the PER. The second factor is dominant with MIMO.

- ***On SISO links, at a given SNR, the PER with M-3P (SF) is still higher compared to the PER with M-3P (FF); however, the impact is less compared to the BER performance:*** The above-mentioned two conflicting factors impact SISO performance as well. However, the second factor (error clustering) is unable to offset the poor performance of FEC codes due to the presence of bursty errors on SISO links.

However, despite including all components of the 802.11 PHY layer, we cannot obtain an exact match between the SNR-PER trend curves in Figure 4.3 and in Figure 4.6(b). In the following, we attribute the behavioral nuance to four characteristics/components of the practical system, which are not implemented in our bit-level simulations for generating M-3P (SF) either due to computational complexity or unpredictability of the behavior.

1. Bursty errors of varying size: M-3P (FF) assumes a uniform distribution of bit errors within a packet; this is based on the premise that due to interleaving/de-interleaving bursty errors are converted into randomly placed errors. In other words, the model implicitly considers an infinite-length interleaver. However, in reality interleaving is done over finite blocks of data. Thus, bursty errors are likely to exist even after interleaving. In fact, as discussed in Section 4.2, our measurements (and other prior work [64]) suggest that a burstiness in errors exists even in packets that are received at the MAC layer. To characterize the impact of bursty errors M-3P (SF) assumes that the fade duration is equal to the length of the packet. As discussed earlier, this impacts the performance of FEC with slow fading; in addition, errors are clustered into fewer packets (rather than being spread out among a large number of packets).

At lower rates, the clustering effect dominates: at these rates the number of bit errors are few, and there are both bursty and isolated bit errors. Although isolated errors are corrected with FEC codes for the most part even with SISO, bursty errors remain. As discussed earlier, MIMO diversity does not help significantly in these settings. Compared to M-3P (SF), the measurements display less frequent burstiness;

consequently, more packets are affected for a given BER. As a result, the measured PER is higher than PER with M-3P (SF).

At the higher rates, there are many bits in error. In particular on SISO links, the FEC codes are unable to correct all bit errors. With MIMO, many of these errors are overcome due to diversity (in these regimes, the gain due to MIMO diversity is more significant). The burstiness of errors is more frequent with M-3P (SF) compared to the measured results. Thus, the FEC codes are more effective in reality than what is seen with M-3P (SF). As a result, the measured results yield a lower PER for a given SNR as compared to the model.

In reality, the burst length of errors is expected to vary, depending on the duration of the fades. Fade duration is an environment-specific factor and it is extremely difficult to measure. Furthermore, the time-correlation between bit-errors can vary temporally. A high-accuracy characterization of the burstiness of errors, to closely reflect what is observed in practice, remains a challenge. Markovian models have been used to characterize bursty errors [5], but in the complex real systems that we attempt to characterize, there may be long range dependence. Note that the Viterbi encoding/decoding itself induces error propagation among bits. As we see later, M-3P (SF) seems to reasonably characterize the performance at the higher layers. Thus, we believe that this model is sufficient for modeling static networks.

2. The use of OFDM: Unlike in the real 802.11n system, M-3P models do not implement an OFDM sub-system. The use of FEC in conjunction with OFDM (as in IEEE 802.11n) gives an additional level of diversity that may boost performance. Potentially we could better conform with measurement results if we could incorporate OFDM into our PHY layer simulations. The slopes of the measured results are much steeper than what is observed with M-3P (SF) (Fig. 4.6); this is likely to be due to additional orders of diversity which are possible with OFDM. However, measuring the frequency selectivity order L (the number of resolvable multipath components) is not an easy task; this is dependent on the environment and could also temporally vary. One could perform simulations with a number of values for L. Due to the very high complexity of simulations with ODFM, we plan to examine these issues in the future.

3. Correlated fading paths: In all of our MIMO models we assume that the path between a pair of antenna elements (one transmit and one receive) experiences a fade that is independent of the fade between any other such pair. This is the basis for most of the reported PHY layer results [39]. However, in an indoor wireless setting, this may not be the case. In other words, the fading would be Ricean [39] as opposed to Rayleigh as assumed in the simulations. Estimating the Ricean factor K [39] in a real system is difficult. Again, incorporating Ricean fading and evaluating the performance for many K values can improve fidelity but at increased simulation costs.

4. Path Loss and Shadowing Effects: In addition to the above factors, the BER vs SNR (or PER vs SNR) curves resulting from the models do not consider path loss or shadowing, whereas the measurement-based curves do. Since the network is static, one might expect these factors to cause long term "fixed" attenuations and to not affect the behavioral results (although the accuracy may be slightly affected). We seek to carefully evaluate these aspects in future work.

4.5 Summary

In this chapter we have examined four models that represent the physical layer behavior of MIMO diversity gain. We have compared these models with our measurements on an 802.11n testbed, in terms of the PER-SNR relation at different rates. We have shown that simplistic models do not conform to measurements in terms of the predicted PER-SNR relation. We have identified the components of the 802.11n physical layer, which are not considered by the simple models, and built new models that incorporate these components; we have shown that more comprehensive models have a higher fidelity to measurements.

Although MIMO products are commercially available, the IEEE 802.11n specification is not a standard yet. Hence, different vendors may implement the protocol differently; future changes may also be forthcoming. Hence, the critical in-depth understanding of how MIMO-STBC links operate, as well as the understanding of cross-layer interactions that will be discussed in the next chapter, will be useful in the design and evaluation of future MIMO-based systems. In the next chapter, we evaluate the *end-to-end* performance benefits due to diversity gain.

Chapter 5

MIMO Diversity Gain in Multi-Hop Networks

In this chapter we evaluate the impact of diversity gain on the higher layer performance in multi-hop networks. From the perspective of the higher layer, we compare the diversity gain impact using each model from Section 4.3 at the link level, to the impact using the measurement-driven benchmark model from Section 4.2. We show that the use of simplistic models can lead to misleading conclusions with regards to the higher layer performance with MIMO diversity. We highlight the following *cross-layer* interactions that affect the end-to-end benefit due to MIMO diversity gain: (a) PHY layer gains due to MIMO diversity do not always carry over to the higher layers, (b) the use of other PHY layer features such as FEC codes significantly influence the gains due to MIMO diversity, and (c) the choice of the routing metric can impact the gains possible with MIMO.

5.1 Introduction

Chapter 4 presented a measurement-driven representation of MIMO-STBC communications, and four physical layer models of these communications. In this chapter, we compare each model with the measurement-based observations, in terms of (i) the throughput increase on a single link and (ii) the end-to-end throughput increase, which are seen as a consequence of the MIMO diversity gain at the physical layer. For this comparison, we integrate the physical layer representation of the models and the measurements in single-hop and multi-hop network simulations. This process reveals the impact of various cross-layer interactions on the performance gains possible with MIMO diversity.

An accurate estimation of the end-to-end throughput hinges on a realistic characterization of the PHY layer. Our study demonstrates that over-simplified physical layer models that have been used in previous studies [86, 41, 29] result in *inaccurate* assessments regarding the higher layer performance. We also show that the throughput benefits due to MIMO diversity are not only affected by other PHY layer features (such as FEC codes), but also by higher layer functions (such as the routing metric). To the best of our knowledge, our work is the first to provide an understanding of how the spatial diversity of MIMO interacts with the different layers of the protocol stack. We present a summary of some of our findings below:

- The link layer performance benefits due to MIMO diversity (as compared to SISO) depend on the transmission rate. In particular, for a desired packet error rate (PER), the signal to noise (SNR) requirement with MIMO decreases by only 1 dB at low rates as compared to SISO (as we see later in Figures 4.2 and 4.6 (c)). This is in stark contrast to what has been reported at the physical layer. Previous studies have shown that at the physical layer there is a significant reduction in the SNR requirement for a desired bit error rate (BER) at all rates [39]. We observe this in our simulations as well; as an example, for a BER of 10^{-3}, due to MIMO diversity, the SNR requirement is reduced by 10 dB or higher (as seen later in Figure 4.5). This contrast between the bit-level and packet-level behaviors is due to complex interactions between packetization, use of forward error correction (FEC) codes and the fading characteristics (discussed later).

- The use of FEC impacts the benefits from MIMO diversity. At low transmission rates, the use of FEC codes are effective in coping with channel induced errors and thus, MIMO diversity is not of much use. At higher transmission rates, MIMO diversity increases the effectiveness of FEC.

- The end-to-end performance benefits of MIMO depend on the choice of the routing metric. While the performance benefits are significant with shortest path routing, the use of a path metric that accounts for link quality, such as ETX [23], diminishes the possible gains. Note that in absolute terms, ETX still provides performance benefits over shortest path routing; however, MIMO diversity provides diminished performance gains over SISO when ETX is used. In our experiments, we find that as compared to shortest path routing, the end-to-end throughput gain is lower by about 15% when ETX is used. In our evaluation, we chose ETX because it is a well understood path metric. Several other path metrics, some shown to perform better than ETX, have also been proposed in existing work. Routing based on these other path metrics is expected to bring up similar issues that we address while studying ETX.

Simulation Parameters		
Channel Parameters	Path Loss Exponent (α)	3
	Stdev.of lognormal shadowing(β)	6
	Thermal Noise	-96 dBm
	Rayleigh fade duration	$\sim$0.02s. for -5 dBm below RMS [71]
MAC Parameters	Receiver sensitivity	-80 dBm
	ECC threshold	1/12000
Traffic Load	5 CBR flows	10 pps/flow
	Packet Size	1500 bytes (12 Kbits)

Table 5.1: Higher Layer Simulation Parameters.

5.2 Models versus Measurements: Comparison Methodology

We implement MDM from Section 4.2 and each of the four models from Section 4.3 on OPNET v. 14 [2]. In this section we describe how the physical layer is modeled and how each model is implemented on OPNET. We explain which operations take place at the MAC and routing layers, and specify the different settings with which each model is examined. The parameters that assist in creating different settings are listed in Table 5.1.

Channel model: OPNET uses a free space path loss propagation model. We implement lognormal shadowing between each pair of nodes. In each setting, the shadow fading attenuation among node pairs is held constant for the duration of the simulation ($\sim$4 minutes); in addition, links have the same shadow fading during the evaluation of each model. Each packet experiences an instantaneous multi-path fade. The SNR versus PER (or BER) curves in Section 4.3 already account for Rayleigh fading, the bit-level simulations implement fast and slow Rayleigh fading channels as discussed earlier. The channel parameters used in the network simulations are listed in Table 5.1.

Packets that are received with a power that is less than the receiver sensitivity are considered as noise. For the remaining packets, the SINR is computed from the reception power of the packet, the thermal noise, and the aggregated power from all interferers. For network simulations, it is extremely difficult, if not impossible, to accurately characterize topology-dependent interference effects. In our work, we assume that the interference also has a Gaussian distribution as the thermal noise; as a result, we map the SINR to the PER/BER. This assumption may cause slight inaccuracy, which however exists for all models being evaluated, including the MDM.

Simulating the models: If M-1 is used, the computed SINR is incremented by the diversity gain from Table 4.1. The resulting SINR is compared to that required for a BER of 10^{-3}; if the value is higher than the requirement, the packet is deemed successful. If M-2 or M-3 (FF) is used, the reception SINR is mapped to a BER from the appropriate BER-SNR curve (using the quantized bins). Random bit errors are then placed within a received frame as per a Bernoulli distribution based on this BER. The packet is deemed successful if the total number of bit errors is zero. We do not simulate M-3 (SF)). To implement this model, we would need to generate bursty errors of the size of a packet (which essentially translates to generating errors at the packet level). A cursory comparison of the link-level performances of M-3 (SF) and MDM indicates significant differences between the diversity gain represented by these two models, especially on SISO links (Figures 4.5(c) and 4.3). Its divergence from our benchmark model MDM at the link-level does not motivate a higher-layer study of M-3 (SF).

With M-3P, we measure the SINR and discard packets at the PER specified by the corresponding PER-SNR curves. As we use the MDM from Section 4.2 as a benchmark for the evaluation of the generated models, we also perform network simulations with MDM. On single hop links, we directly compare the performance of our models with the measured throughput from our testbed. In multi-hop settings, we use MDM as we use the other models, since the IEEE 802.11n draft does not yet support the ad-hoc mode [84]. In these simulations, we discard packets as per the PER in Fig. 4.3.

We note that the SNR-PER (or SNR-BER) trend curves in Figures 4.6 (and 4.5) are plotted only for a range of SNR values. In the simulations, the SNR value could be higher than the maximum plotted, while the corresponding PER (or BER) values may not be directly available from these plots. When this occurs, we perform a linear extrapolation to compute the appropriate PER (or the corresponding BER values).

MAC layer: At the MAC layer we use the DCF mode of 802.11. We use the SIFS, minimum backoff window and data rate values as specified in the 802.11g standard and the 802.11n draft while simulating the SISO and MIMO links, respectively. In the MIMO scenario, the Reduced Inter-frame Spacing (RIFS) equivalent of 2μs is used instead of the 10μs SIFS interval of the SISO setting. We do not use RTS/CTS messages, as this is a common design decision in practice [23]. In order to examine the impact of diversity gain with MIMO at the different rates of operation (1 bps/Hz, 2 bps/Hz and 3 bps/Hz), we fix the transmission rate to the desired value. Note that in practice, a rate adaptation mechanism is typically used. Our results apply with rate adaptation mechanisms; at each transmission rate, the rate-specific gain with MIMO will be achieved based on the SNR.

Routing in the multi-hop scenarios: At the routing layer we use the popular Dynamic Source Routing (DSR) protocol [43]. DSR has also been used in prior work for studying routing on MIMO links [86]. For the route discovery control packets we use the same fixed rate as for data packets; this is in order to ensure that the discovered route is sustained for data communications at that rate.

We simulate scenarios when DSR chooses the minimum-hop routes or minimum-ETX [23] routes. The first policy refers to the DSR version that uses hop count as the routing metric. In this policy, DSR chooses the minimum-hop route from among those that are discovered. When this route fails, other cached routes may be used (we examine DSR without caching in 5.3.2). With the second policy, we compute the ETX metric to be the inverse of the link's reliability. In particular, ETX is quantified in terms of the packet delivery rate (PDR) on the link for the corresponding rate [23]. With the bit-level abstractions, we compute this value off-line assuming independent errors. With minimum-ETX routing, intermediate nodes are not allowed to report cached routes to the source as in the default ETX implementation.

Topologies: We first consider **single-hop** flows to compare the link throughput from our testbed with those achieved by the models on the simulator. The goal of this study is to evaluate the simulator's reliability in terms of providing realistic multi-hop realization of MDM as a benchmark, and assessing the fidelity of the considered models. In these simulations, we vary the SNR on the considered links by adjusting miscellaneous channel parameters (such as the distance and shadowing). We consider these links in isolation; thus, the only noise is the thermal noise. Next, we consider randomly generated **multi-hop** topologies of different densities, where nodes are distributed in a 1000m×1000m area. On this area of deployment we consider networks of different density. We perform two sets of simulations on multi-hop topologies; with multiple and single flows. In the first set, five UDP flows are generated between random source-destination pairs at a rate specified in Table 5.1. We vary the source and destination nodes in order to simulate for different end-to-end hop-distances. Each scenario is simulated ten times; thus, each scenario is evaluated with fifty different source-destination pairs. In the second set of simulations, we study a single flow in order to reduce route failures due to interference, and to capture the impact of the routing metric on the network-level gains due to MIMO diversity. In these simulations, we modify routing policies to disallow the reporting of cached routes.

Generality of our results: The measured results are from a static indoor wireless network. In such a network, slow fading would be a good model (as demonstrated by our studies). However, depending on the network deployment, appropriate models will have to be used. In a rapidly changing environment or with high mobility, the fast fading model is likely to be more appropriate (study of such a system will be considered in the future). If there are direct line of sight links (such as in an open field) the multi-path fading models may be completely invalid and may have to be replaced with simple path loss models. In fact, in such cases, MIMO may not offer any benefits. At even higher rates, the gains due to MIMO diversity may be more prominent. Finally, we wish to point out that the PER results that we provide are for a fixed packet length of 1500 bytes. The PHY simulations will have to be re-run if we had packets of varying sizes. One way of making the results more general is to classify packets into different types (short, medium long) and have PER rates for the average packet size in each class. Our studies provide a rich set of results that can be applied in a large set of possible scenarios (although not exhaustive).

On the choice of the transmission rates: We primarily consider the bit rates of 1-3 bps/Hz. It is evident from our work that at low rates MIMO does not offer significant benefits. As a result, with transmission rates lower than 18 Mbps (1 bps/Hz), we would not see any gain. On the other hand, 54 Mbps (3 bps/Hz) is the highest possible transmission rate with SISO. At even higher rates, one might expect the gains due to MIMO diversity to be more pronounced. For a fair comparison, we choose this to be the highest rate that we experiment with. Finally, we pick 36 Mbps (2 bps/Hz), since it is the median rate between 18 and 54 Mbps.

More complex MIMO systems: While the work here focuses on MIMO diversity and 2x2 systems, one might expect that similar issues will exist with higher orders of diversity, multi-user MIMO systems or with space division multiplexing. Obtaining a deeper understanding and developing high fidelity models for these systems is a challenge.

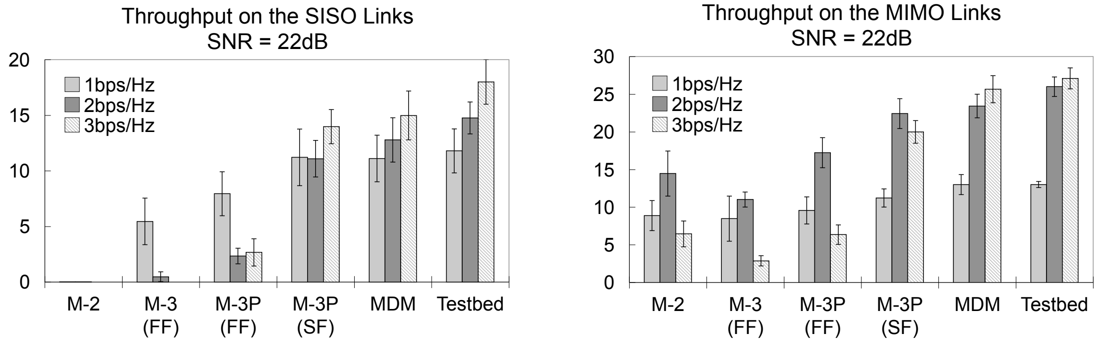
5.3 Simulation Results and Discussions

In this section we first present the throughput performance with all models (including MDM) in single-hop settings, and compare these with the throughput measured on our testbed. Next, we compare the network-level performance with all models on multi-hop networks, for the cases of a single flow and multiple flows. Our key observations are:

- The bit-level models almost always overestimate the possible performance improvement with MIMO, and
- Both the performance improvement and the absolute SISO and MIMO performances using the packet-level model on the slow fading channel conform to a large extent with those observed using MDM.

5.3.1 Results from Single-Hop settings

We compare the link throughput obtained in our simulations with those measured on our 802.11n testbed. Our motivation is to find a close match between the throughput measured on the testbed, and the simulated link throughputs. This result supports the reliability of simulations for the assessment on multi-hop topologies.



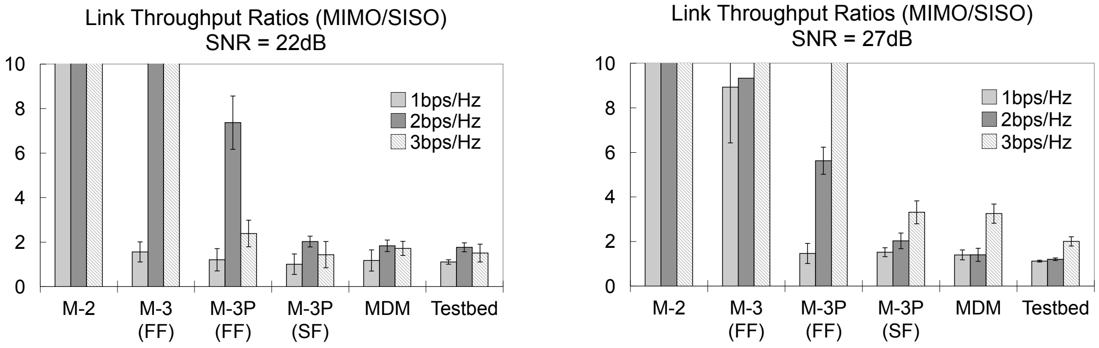
(a) SISO Throughput at 22dB

(b) MIMO Throughput at 22dB

Figure 5.1: A comparison of the link throughput measured on the testbed versus simulated on single-hop topologies, at two selected SNR values.

We measure the SISO and MIMO throughput on different links in our testbed, at the three rates under study. We also simulate single-hop topologies and obtain the throughput on SISO and MIMO links for a range of SNR values at these rates. In Figure 5.1 we present the absolute throughput on SISO and MIMO links at two sample SNR values. We observe that the simulated throughput with MDM differs by at most 11% from what we measure on our testbed. We also observe that the throughput with M-3P (SF) differs by at most 21% from the throughput with MDM, and 24% from the throughput on the testbed (at 3 bps/Hz data rate). On the other hand, the use of M-2 can lead to a drastic overestimation in the performance improvement with MIMO; the absolute throughput is much lower (77% lower with MIMO and 3,000 times lower with SISO) than what is observed on the testbed. Since M-2 does not account for FEC codes, the packet transmissions with this model experience higher losses leading to a poorer throughput; this effect is more pronounced with SISO than with MIMO and thus, the throughput increase with MIMO diversity is extremely high (Figure 5.2(a)). With the fast fading models M-3(FF) and M-3P(FF), the throughput increase with MIMO compared to SISO is again higher compared to MDM (the ratio of the MIMO over SISO throughputs are shown in Figure 5.2). This is because the errors are independent and identically distributed. While the FEC codes are more effective on a fast fading channel, the errors are spread over many packets, leading to poorer throughput in packet-level simulations. With M-3P (SF), the impact of the environment in the indoor setting (bursty errors) is better captured; thus, the performance with this model is similar to the performance with MDM and to what is observed on the testbed.

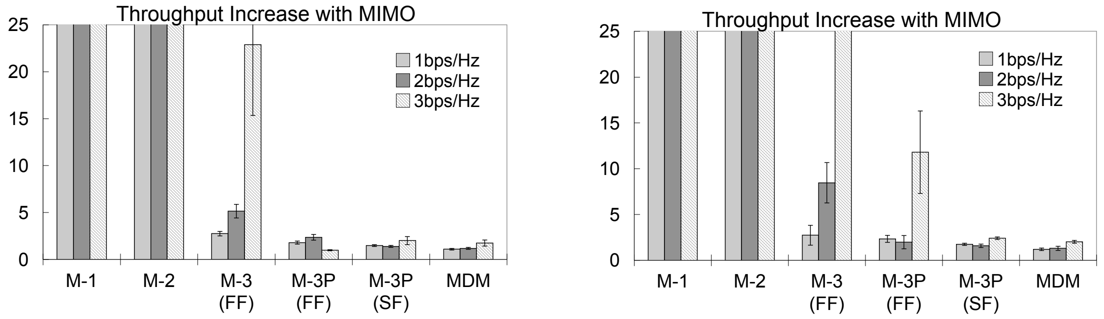
This experiment justifies the effectiveness of MDM as an indicator of what might be possible (a benchmark) in multi-hop settings. The results with simulations are not exactly identical to those with measurements, which is because the wireless channel in reality differs from what is implemented in the simulator. The propagation characteristics with 802.11n are still under study [7] and it is a hard task to implement the channel impact on the bits in a packet in network simulations.



(a) Throughput Ratio at 22dB

(b) Throughput Ratio at 27dB

Figure 5.2: The impact of MIMO diversity gain on the link throughput at two selected SNR values. The figure shows the ratio of the throughput on MIMO links that that on SISO links. A higher ratio indicates a higher diversity gain on a single link.



(a) 50 nodes/1km^2

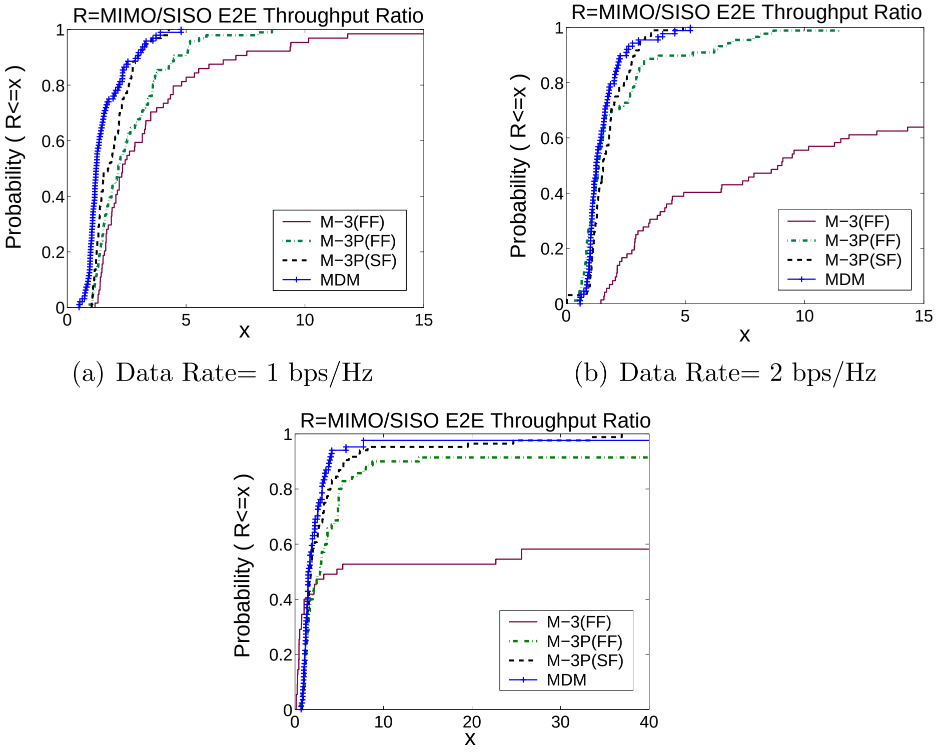
(b) 80 nodes/1km^2

Figure 5.3: Evaluating the impact MIMO diversity gain in multi-hop settings: each bar represents the ratio of the end-to-end MIMO and SISO throughputs using a particular model at a given rate.

5.3.2 Results from Multi-hop Settings

We compare the end-to-end performance using each of the MIMO models from Section 4.3 with that using the MDM from Section 4.2, in multi-hop networks. Our goal is to compare the impact of diversity gain on end-to-end performance using each model for representing the SISO and MIMO links at a given rate. Our study leads to the following observations.

1. Simply using the bit-level models for representing MIMO diversity gain may lead to erroneous conclusions from the perspective of the higher layers. We compute the ratio of the achievable end-to-end network throughputs using MIMO links and SISO links. In Fig. 5.3, we show these ratios in two networks having different node densities. The standard deviations among the samples collected for a given setting (using a particular model and data rate) are shown as error bars. As with the single hop case, all models but M-3P (SF) *overestimate* the end-to-end performance improvement with MIMO compared to SISO, irrespective of the node density. In particular, the end-to-end throughput increase with M-1 and M-2 is as high as 180 times of what is achieved with MDA (for clarity of representation, we truncate these results in Fig. 5.3). Furthermore, both fast fading models M-3 (FF) and M-3P (FF) overestimate the benefits due to MIMO, especially at the high rates (by 29 times and 13 times respectively, at 3 bps/Hz). M-3P (SF) shows a good degree of conformance to the results with MDM; in all simulated densities the throughput ratio with this model differs by at most 19% from the throughput ratio with MDM. The cause of these results is similar to that in the single-hop case.



(a) Data Rate= 1 bps/Hz

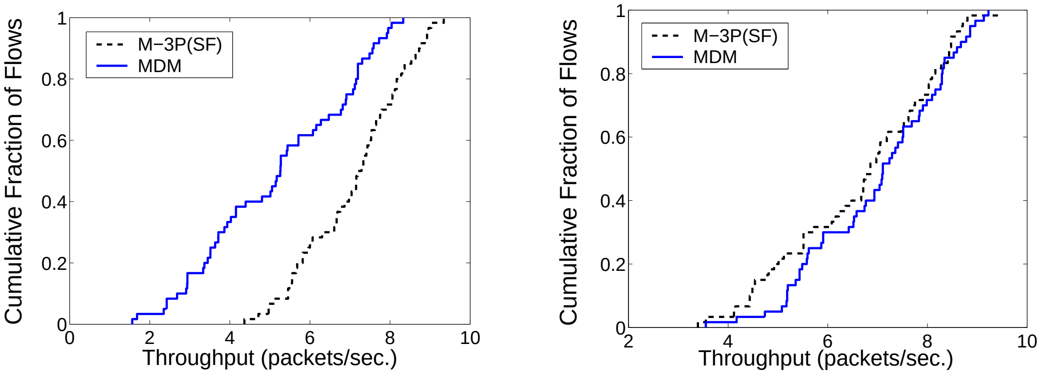
(b) Data Rate= 2 bps/Hz

(c) Data Rate= 3 bps/Hz

Figure 5.4: Cumulative distribution of the per-flow throughput ratio (MIMO over SISO) at different rates, using shortest paths.

In Figure 5.4 we plot the CDF of the MIMO/SISO end-to-end throughput ratio, measured on the flows (*i.e.,* source-destination pairs) in the considered 50-node topologies. We denote this ratio by R; the abscissa x represents the value of this ratio and the ordinate represents the probability that $R \leq x$.

In Section 4.4 we have discussed that the bit-level models M-1, M-2 and M-3 as well as the fast-fading packet-level model M-3P (FF) fail to conform with MDM in terms of representing diversity gain on a link (depicted by the SNR-BER or SNR-PER trend curves). In this section, Figures 5.3 and 5.4 indicate that these models also fail to conform to MDM in terms of their quantification of the network performance. Therefore, we exclude the performance results with these models from the following experiment that compares the MIMO throughput and delay performances of measurements and models.



(a) Throughput at 1 bps/Hz

(b) Throughput at 3 bps/Hz

Figure 5.5: Distribution of the average end-to-end throughput achieved on MIMO links by the flows at the low and high rates.

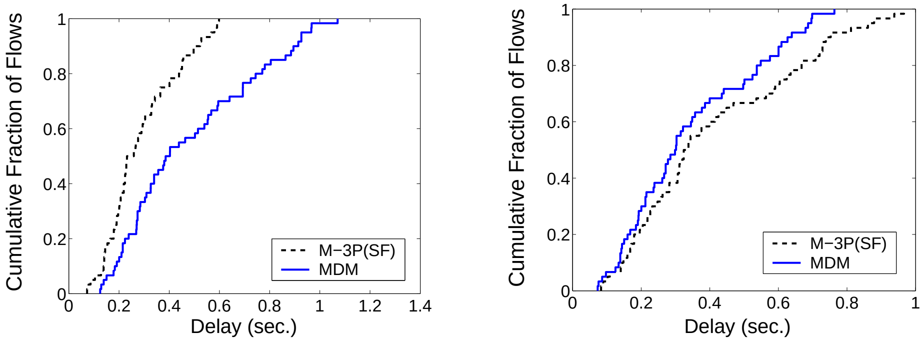
2. The absolute throughput on MIMO links with M-3P (SF) exhibits a high degree of conformance to that with MDM. In Fig. 5.5, we plot the CDF of the end-to-end throughput using MIMO links at the lowest and highest rates. M-3P (SF) yields higher throughput at the lowest rate, and the opposite is true at the highest rate. We observe that both models use routes of similar lengths (see Table 5.2); however, the number of retransmissions incurred using these two models differ at different rates. At the lower rate, links experience more retransmissions with MDM, while the opposite is true at the higher rate. This is because the SNR requirement for a target PER is higher with MDM than with M-3P (SF) at the lower rates (as seen from a comparison of Fig. 4.6(b) and 4.3). Consequently, long links that are established with MDM have poorer reliability than those with M-3P (SF). The opposite effect is seen at the highest rate of 3 bps/Hz. At this rate, the SNR requirement for a target PER is lower with MDM than with M-3P (SF).

	M-3P (SF)			**MDM**		
	1 bps/Hz	**2 bps/Hz**	**3 bps/Hz**	**1 bps/Hz**	**2 bps/Hz**	**3 bps/Hz**
Hop Count (network avg.)	5.67	5.82	5.50	5.35	5.71	5.41
Re-TX. Count (network avg.)	334.76	301.44	684.16	523.08	364.24	407.94

Table 5.2: Statistics regarding the flows whose performance is shown in Fig. 5.5.

Table 5.2 shows that the average number of hops with 2 bps/Hz (or even 1 bps/Hz) is higher than that with 3 bps/Hz. This is counter-intuitive; one would expect the average hop count to increase with rate. Upon close inspection, we observe that this is an artifact of using the corresponding rate for route query packets. There are two conflicting factors when a route query is transmitted at a high rate. On the one hand, the queries are less reliably transported across each hop. On the other hand, each query transmission takes a shorter time thereby reducing interference on other query retransmissions. While the absolute shortest path discovered with 1 bps/Hz is shorter than (or in some cases equal to) the one discovered with 3 bps/Hz (or 2 bps/Hz) due to the former effect, there are fewer cached routes with 1 bps/Hz due to the latter effect. Some of the additional cached paths when using the higher rates bias the average hop count to a value lower than that with 1 bps/Hz. Recall that we transmit route discovery packets at the same rate as the data packets. We more closely examine the impact of route caching on a single flow later in this section.

3. The average end-to-end delay with M-3P (SF) exhibits a high degree of conformance with that using MDM. In Fig. 5.6(a) and 5.6(b) we



(a) Delay at 1 bps/Hz

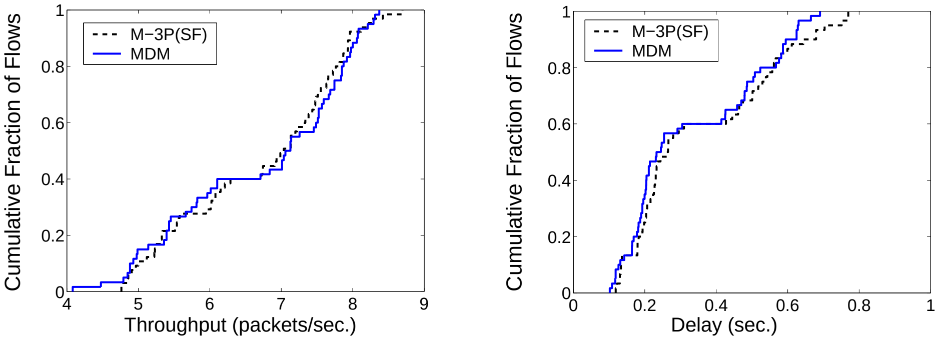
(b) Delay at 3 bps/Hz

Figure 5.6: Distribution of the average end-to-end delay achieved on MIMO links by the flows at the low and high rates.

plot the the CDF of the average delays with M-3P (SF) and MDM at the highest and lowest rates. The cause of the slight differences follow from our discussion earlier. To re-iterate, at 1 bps/Hz, M-3P projects a lower end-to-end delay compared to MDM due to a slightly greater diversity gain. The opposite is observed at the highest rate. The end-to-end delay performance with these two models conform best at 2bps/Hz (we omit this result due to space constraints).

4. The most pronounced benefits with diversity are seen at the highest rate (Fig. 5.3). At the lower rates, the errors are few and the FEC codes mostly correct these errors. MIMO diversity does not provide significant *additional* benefits. However, at the higher rate, the errors are many and the FEC codes are not as effective with SISO. MIMO diversity reduces the extent of errors. The FEC codes are now more effective and thus, a significant gain is observed as compared to SISO.

5. The conformance of M-3P (SF) to MDM increases when minimum-ETX routes are used. This is because ETX links operate at relatively high SNRs where the PER with M-3P (SF) is closer to that with measurements (see Fig. 4.3). In Figure 5.7 we show the CDF of the throughput and delay on MIMO links at 3 bps/Hz. We observe that the two curves corresponding to M-3P(SF) and MDM are



(a) Throughput at 3bps/Hz

(b) Delay at 3bps/Hz

Figure 5.7: The conformance of M-3P (SF) to MDM increases when minimum-ETX routes are used.

closer to each other in this figure, compared to those in Figures 5.5(b) and 5.6(b). This is because ETX links operate at relatively high SNRs where the PER with M-3P (SF) is closer to that with measurements (see Fig. 4.3).

6. Performance improvements due to MIMO relative to SISO decrease with ETX-based routing. On the same 50-node topologies considered with the shortest-path routing, we examine the ratio of the achievable end-to-end network throughput using MIMO links to that using SISO links, with ETX based routing. We observe that with ETX, this ratio decreases compared to the shortest-path routing. We define D to be the reduction in this ratio compared to minimum hop routing; D is depicted in Figure 5.3 for different data rates for MDM and M-3P (SF). We observe that M-3P (SF) shows

As depicted in Fig. 5.8, the gains due to MIMO decrease with the use of ETX by as much as 20%. With ETX, the use of the more stable (albeit longer) routes with more reliable links (higher average SNR) improves the SISO performance as compared to minimum-hop routes. With the higher average SNRs on the SISO links, the FEC codes are more effective and MIMO does not provide significant *additional* benefits.

7. The impact of caching causes a disadvantage of the low-rate schemes

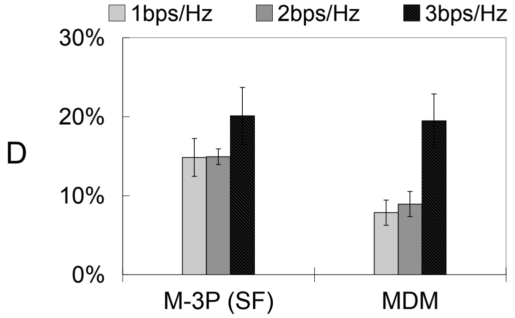
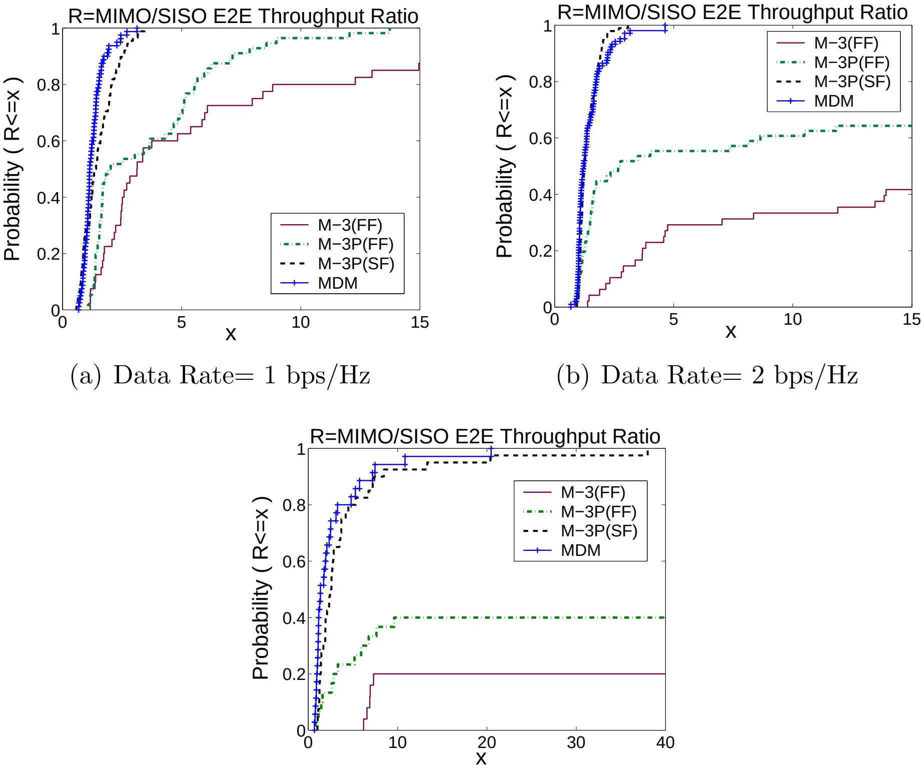


Figure 5.8: Performance improvements due to MIMO diversity gain slightly decrease with minimum-ETX routing compared to the case of shortest routes.

with both MDM and M-3P (SF). The case of multiple flows may hinder (or cause) the observation of certain behavior due to interference-induced link failures. We study flows in isolation, in order to reduce the extent of such route failures. In this controlled study, we also modify the DSR implementation to eliminate the impact of caching on route selection. With route caching in DSR, any node can respond to a route query based on the routes in their cache [43]. This can cause the minimum hop version to choose much longer paths than the actual shortest path. In our controlled scenario, we disable this feature and only the destination can respond to RREQs. We consider 30 scenarios and pick 7 scenarios where the source and the destination pairs are separated by a distance of approximately 5 hops (the results were similar for other pairs that were separated by different distances). In general, we find that the minimum-ETX routes are slightly longer (but more reliable) than the shortest paths on average, with both SISO and MIMO. The PDRs with SISO increase on such routes and thus, the gains due to MIMO are reduced. As a direct consequence, the MIMO to SISO throughput ratios are smaller using minimum-ETX routes compared to minimum-hop routes. We do not provide detailed results due to space constraints. However, in a nutshell, we find that at 3 bps/Hz, with M-3P (SF) the average route



(a) Data Rate= 1 bps/Hz

(b) Data Rate= 2 bps/Hz

(c) Data Rate= 3 bps/Hz

Figure 5.9: Cumulative distribution of the per-flow throughput ratio (MIMO over SISO) at different rates, using minimum-ETX routes. The MIMO/SISO performance improvements become less visible when minimum-ETX routes are used.

length in hops with SISO (MIMO) increases from 6.6 (5.9) to 6.8 (6.3) with the use of ETX. Correspondingly, for this rate, the gain due to MIMO reduces from 28% to 9%, relative to SISO. The observations are similar with MDM. The MIMO gain reduces from about 35% to about 7%, relative to SISO; the route length in hops increases with SISO (MIMO) from 6.4 (5.9) to 6.8 (6.3) due to ETX.

5.4 Conclusions

In this chapter, we studied the impact of MIMO diversity on the higher layers, considering the various dependencies between the PHY, MAC and routing layers that influence the gains possible due to diversity from an application throughput perspective. The study is based on comparing the measured performance on an indoor 802.11n testbed with performance derived from a set of PHY layer models of varying complexity. This comparison is assessed in network simulations that incorporate different physical layer models for representing the communications between node pairs.

Our study reveals that the diversity gain at the bit-level does not always appear at the packet level. Furthermore, when MIMO is used in conjunction with other PHY layer features such as FEC codes, it may not offer benefits at certain rates. Finally, we find that the routing metric used may impact the gains possible with MIMO. Our study can be useful in the design of higher layer protocols for use with spatial diversity; as an example, the behaviors of MIMO at different rates might influence the way in which the rate adaptation algorithm is designed. In addition, it may lead to the design of new routing metrics that jointly consider rate and the gains that are possible with diversity. We leave the exploration of these possibilities as future work.

Chapter 6

Multi-User MIMO Receptions in Multi-Hop Networks

With antenna arrays, nodes can decode multiple concurrent signals using Successive Interference Cancellations (SIC). Multi-User MIMO receptions refer to the case when the concurrent signals are the transmissions of different nodes. SIC-based receptions are successful only if each signal satisfies a signal-to-noise-plus-interference ratio (SINR) requirement at a receiver. Consequently, only a carefully selected subset of the communications in a multi-hop network can be carried out successfully at any given time. In this chapter, first, we specify the criteria for a proper selection of communication links that facilitates successful decoding at the receivers. Second, we formulate the problem of dividing the communications in a multi-hop network into a small number of *sub-topologies*, such that the decoding criteria is met in each of these sub-topologies. Finally, we examine the hardness of this problem and delineate a solution approach.

6.1 Introduction

Multi-user MIMO communications are facilitated at the PHY (physical) layer in two different ways; (a) Multi-user MIMO transmissions using precoding at the transmitters (with dirty paper codes [69]), and/or (b) Multi-user MIMO receptions using layered decoding at the receivers (with SIC as proposed in the V-BLAST architecture [104]). Precoding requires transmitters to have an accurate estimate of the channel realization at the receivers. Due to the prohibitive transmission overhead needed for the feedback of this information from all potential receivers to a transmitter, precoding is not widely employed in practice [76]. On the other hand, SIC only requires the knowledge of channel estimates at the receivers; since this can be provided in practice by the transmission of pilot tones [106], SIC is considered to be a more practical approach for multi-user MIMO systems.

SIC can enable the decoding of weak signals in the presence of strong interference: With SIC, the receiver tries to decode multiple received signals, typically each coming from a different transmitter[1], using an iterative approach. In each iteration, the strongest signal is decoded, by treating the remaining signals as interference. If a required SINR is satisfied, this signal can be decoded and *removed* from the received composite signal. In the subsequent iteration, the next strongest signal is decoded, and the process continues until either all the signals are decoded or a point is reached where an iteration fails. Note that after the removal of the strongest signal in an iteration, fewer interfering signals remain for the next iteration. With this process, nodes can receive signals from weaker transmitters in the presence of strong interferers. To illustrate with an example, consider that Jack wishes to de-

[1] SIC can be potentially to decode multiple signals from the same transmitter. However, this is not practical, since such signals are received with similar powers.

code Chloe's signal, in the presence of a stronger interfering signal that comes from Nina. Jack needs to first decode Nina's signal correctly, since it is the stronger one, treating Chloe's signal as noise. If the SINR is satisfied and Nina's signal is decoded in the 1st iteration, Jack can proceed with removing it from the composite signal and decoding Chloe's signal in the 2nd iteration. SIC can be cojoint with *selection diversity* at the transmitters. To facilitate this, Jack sends feedback to Chloe with regards to the channel attenuation experienced by Chloe's pilot tones, transmitted with different antennas. Chloe then uses the antenna element that yields the best reception at Jack's end [34], as we discuss later.

Challenges in rendering SIC efficient: There are two challenges that arise in harnessing the capacity benefits with SIC in a multi-hop wireless network. First, only up to a certain maximum number of strong interferers can be removed from a composite signal. This number is equal to $A-1$ where A is the number of antenna elements at the receiver. Second, the decoding in each successive iteration is successful only when signals are received with disparate power levels [94]. This is evident from the above example; indeed, if Chloe and Nina's signals both arrive at Jack with the same signal strength, the SINR requirement for decoding Nina's signal is unlikely to be satisfied and SIC will fail. The challenge here is to impose some form of control such that only links that allow SIC to succeed with high probability are active simultaneously. Previous studies on realizing multi-user MIMO communications try to simply limit the number of links that simultaneously access the medium [22, 87, 65]. However, they do not account for the reception powers of the signals. In most of these studies, the communication model allows as many concurrent transmissions as the number of antenna elements in each contention region.

Addressing the challenges: In this chapter, we ask the question: *How can we construct an environment that is conducive to SIC, in terms of a clean SINR separation between the target and the interfering signals?* We propose a solution approach that bounds the "strong" and "weak" interference on every communication in order to facilitate the SIC functionality. Strong interference terms are the transmissions that are received with a higher signal strength than the target transmission, and weak interference terms are weaker than the target signal. Our goal is to identify sets of links (sub-topologies), such that:

- In each sub-topology, any simultaneously active set of transmitters construct signals with sufficiently disparate reception powers at every target receiver. Hence, these receivers are able to filter out stronger signals and detect the desired data stream(s) using SIC.
- The links that cannot successfully co-exist are included in different sub-topologies and thus, they are activated at different times.
- The union of these sub-topologies spans all the links in the multihop network.

Note that there is a trade-off between the medium-access delay experienced by a link and the probability of successful reception. For each link to obtain frequent medium access, few sub-topologies should be constructed, each of which is packed with as many links as possible. On the other hand, a larger number of active links decreases the probability of successful detection with SIC. Our work is the first to consider this trade-off towards enabling multi-user MIMO communications.

In this chapter we state our goals, develop our models and thus postulate the MU-SIC problem. We show that finding the optimum solution to MU-SIC is NP-hard.

We thus provide heuristics for a solution approach. The design and evaluation of heuristic-based algorithms will constitute Chapter 7.

6.2 MU-SIC Problem Formulation

Graph Model of the Network: We model the multi-hop network as a directional graph G=(V, E). The set V includes all the nodes in the network. For a pair of nodes u,$v \in V$, a directional edge e=(u, v) is a member of E if node u's signals can be decoded at v in the absence of interference.

$$\frac{\mathrm{E}\left[P_{RX}(u, v)\right]}{N} > \gamma \tag{6.1}$$

The channel model: Transmitted signals are attenuated due to path loss. Furthermore, due to multipath, the signal experiences a fade that varies as per a Rayleigh-distribution; the channel coefficients $h_{u,v}$ in Eqn. 6.2 are generated using this model. The fading is independent among different transmitter-receiver antenna pairs, as well as at different time instances. Given these, the power of a signal transmitted by node u, at a receiver node v is given by:

$$P_{uv} = \frac{P \cdot |h_{uv}|^2}{d_{uv}^{\alpha}} \quad . \tag{6.2}$$

Multi-User MIMO Receptions using SIC: With SIC, the communication on a link (u, v) is successful **iff**:

(i) At v, signals from no more than A-1 other transmitters are received with a power that is greater than that of u's signal, **and** each stronger signal is successfully *removed* before decoding u's signal. For this, in each of the j={1,2,...A-1} SIC iterations, the following must be satisfied for detecting the j^{th} strongest transmitter's signal:

$$\frac{P_{uv}}{N + \sum_{z \neq u} P_{z,v}} > \gamma \tag{6.3}$$

(ii) After removing all stronger interferers, Eqn. 6.3 must be satisfied for u's signal.

In order to satisfy these criteria, we need to partition the network into sub-topologies that time-share the medium. In each sub-topology, an active link has to satisfy Eqn. 6.3 with an arbitrarily small probability of decoding error δ. Note that the condition in Eqn. 6.3 is only probabilistically satisfied since temporal channel fluctuations can cause the decoding to fail. Our goal is to minimize the number of such topologies in order to ensure that each link is activated with the maximum frequency. Therefore, indirectly we seek to maximize the throughput under the conditions of max-min fairness.

The two objectives of minimizing the number of sub-topologies and minimizing the probability of decoding error at the receivers conflict due to interference among the links in the network. Therefore, to identify the links that can be grouped under the same sub-topology, we first examine the interference among them. Interference among link pairs has been represented in an *interference graph* (or conflict graph) in the context of single-antenna networks in [17, 40]. In single-antenna networks, the interference graph is preferably represented among node pairs as opposed to link pairs, due to the complexity of link-based representation [17]. In our work, however, the interference among communications is not only determined based on whether a node is transmitting or not, but also on the receivers targeted by each transmitter. The antenna element that is used for transmission varies based on the receiver due to MIMO selective diversity, and consequently the interference that is projected on other neighbors also varies. In the following we introduce the interference graph G'

of G and formulate the MU-SIC problem on G'.

6.2.1 The Interference Graph

The directional, edge- and vertex-weighted interference graph $G'{=}(V', E', w_{V'}, w_{E'})$ is induced from the directional graph $G{=}(V, E)$ that represents the network as follows. V' includes a vertex e for each directional edge (u, v) in E. The vertex e is weighted by the *mean* value of P_{uv} (Eqn. 6.2); the weight of edge e is denoted by $w_{V'}(e)$. Each pair of vertices e,f in V' is connected by a weighted directional edge in E'; the weight of the edge (e, f), denoted by $w_{E'}(e, f)$ is equal to the power measured at the receiver of link f when link e is active.

The problem of dividing the graph G' into a minimum number of sub-graphs G'_i (to represent each sub-topology) can be formally represented as follows:

minimize $\quad m, \;$ where $V' = V'_1 \cup V'_2 \cup .... \cup V'_m$

subject to $\quad \forall i, \, \forall e \in V'_i :$

$$\sum_{g \in V'_i} b_g < A \;\; , \quad \text{where} \quad b_g = \left\{ \begin{smallmatrix} 1 & , & w_{E'}(g,e) \geq w_{V'}(e) \\ 0 & , & otherwise \end{smallmatrix} \right\} \tag{6.4}$$

$$\forall_{\substack{g \in V'_i \quad s.t. \\ w_{E'}(g,e) \geq w_{V'}(e)}} \left[\sum_{\substack{h \in V'_i \\ w_{E'}(h,e) < w_{E'}(g,e)}} w_{E'}(h, e) < \frac{w_{E'}(g, e)}{\gamma} - N \right] \tag{6.5}$$

$$\sum_{\substack{f \in V'_i \\ w_{E'}(f,e) < w_{V'}(e)}} w_{E'}(f, e) < \frac{w_{V'}(e)}{\gamma} - N \tag{6.6}$$

This requires the minimization of the number of sub-topologies m. V'_j represents the set of links in the j^{th} sub-topology that is constructed. The constraint in inequality 6.4 mandates that the total number of strong interferers[2] should be less than the degrees of freedom A. The constraint in inequality 6.5 requires that for each communication link in a group, **each strong** interferer must be correctly decodable to be removed; the term within the summation inside the parentheses, essentially implies that the SINR for each strong interferer should be satisfied. Inequality 6.6 mandates that once the interference from the strong interferers is removed, the desired signal should be decodable i.e., should satisfy the SINR requirement. Note that the latter two inequalities are easily obtainable from the requirement postulated in Eqn. 6.3.

6.2.2 The Complexity of the MU-SIC Problem

We prove that the problem of finding a minimum set of sub-topologies such that SIC is successful, is NP-hard. To this end, we simply show that a special case of the considered problem is the problem of scheduling in a SISO network.

Theorem 6.2.1 *The topology control problem for multi-user MIMO networks using SIC, based on the SINR model is NP-hard.*

Proof 6.2.2 *When $A = 1$ in Eqn. 6.4, the considered problem reduces to the SISO scheduling problem with a single antenna per device. In this context, no SIC is considered. The SISO scheduling problem has been proven to be NP-hard [31]. Since, a special case of the considered problem is NP-hard, we conclude that the general problem is NP-hard.*

[2] If the incoming edge weight $w_{E'}(g, e)$, at a node e of the interference graph, is larger than node weight $w_{V'}(e)$, link g projects interference that is stronger than the desired signal strength at the receiver of link e.

6.3 Proposed Solution Approach to MU-SIC

Our approach relies on controlling the interference on each link.

6.3.1 Limiting Weak Interference

In order for a signal to be decoded in the presence of only weak interferers, Eqn. 6.3 has to be satisfied for this target signal (*i.e.,* u's signal at node v). From this equation it is easy to see that the maximum interference that can be tolerated on link (u, v) for successful decoding under these conditions is $P_{uv}/\gamma - N$. We refer to this value as the *weak interference budget* of link (u, v). At the receiver node v, the aggregate power from all the weak interferers is the sum of the P_{uv} terms as defined in Eqn. 6.2. The $|h_{zv}|^2$ terms in this sum are exponentially distributed random variables (h_{zv} follows a Rayleigh distribution). Thus, the aggregate interference from the transmitters *at a given distance* can be modeled as an Erlang-distributed random variable with parameters n, the number of these transmitters, and σ, the variance of the Rayleigh distributed random variable. Our goal is to ensure that the aggregate weak interference on each given link is smaller than its weak interference budget with a probability greater than $1 - \delta$; in other words, the probability that the weak interference budget is exceeded is bounded by an arbitrarily small value of δ:

$$Pr\left\{\sum_{z\neq u} P(z, v) > \frac{P(u, v)}{\gamma} - N\right\} < \delta. \tag{6.7}$$

Assuming that all weak interferers are received with the same power as that of the strongest (a conservative condition) among these interferers and using the cumulative distribution function (CDF) of the Erlang-distribution, we define a function

$ErlangFeasible(\sigma, n)$ to bound the weak interference (n is the total number of weak interferers). This function outputs the tuple (n_{max},P'_{max}), where n is the maximum number of interferers that can be tolerated by the target link, if each of these n interferers projected an interference equal to P'_{max}. If $n_{max} \neq n$, the strongest weak interferer is removed from consideration, and $ErlangFeasible(\sigma, n-1)$ is computed for the next strongest weak interference power. Thus, the $ErlangFeasible()$ reduces Eqn. 6.7 to:

$$Pr\left\{ErlangFeasible(\sigma, n) > \frac{\frac{P(u,v)}{\gamma} - N}{P_{max}}\right\} < \delta. \tag{6.8}$$

With MUSIC, this condition is used to bound the number of weak interferers for a link.

6.3.2 Controlling Strong Interference

Consider an example where a node receives a composite signal with K different signals. Further, assume that the signal from its desired transmitter is the second *weakest* from among these K signals. P_i represents the power of the i^{th} strongest signal in a sorted order; thus, $P_{(K-1)}$ represents the power of the desired signal. In order for the desired signal to be correctly decoded, first, it is required that $K \leq A$. Second, each of the $K-2$ strong interference terms must be correctly decoded and removed. This implies that the SINR requirements at each of the following steps need to be satisfied:

$$\begin{aligned}
&\text{if}\quad P_1/(N+P_2+P_3+...+P_K) > \gamma \Rightarrow \text{decode and subtract 1st interferer.}\\
&\text{if}\quad P_2/(N+P_3+...+P_K) > \gamma \Rightarrow \text{decode and subtract 2nd interferer.}\\
&\qquad \cdots\\
&\text{if}\quad P_{K-1}/(N+P_K) > \gamma \Rightarrow \text{desired transmitter is correctly decoded.}
\end{aligned} \tag{6.9}$$

In the set of requirements listed above, the strong interference P_{K-2} must be at *least* equal to $(P_{k-1}+P_K+N).\gamma$. This condition has to be satisfied in the worst case when P_K has its maximum value of $(P_{K-1}/\gamma)-N$. Simplifying, it is easy to see that $P_{(K-2)}$ must be at least equal to $P_{(K-1)}\cdot(\gamma+1)$. Following through these equations iteratively backwards, each j^{th} strongest interferer must satisfy the property relative to the intended signal:

$$P_j \geq P_{(K-1)}\cdot(\gamma+1)^{K-1-j}, \forall j \in [1, K-2] \tag{6.10}$$

For correct decoding at every step, there can be at most one strong interferer for a specific value of $l = K-1+j$. In other words, if a link (y,z) experiences interference from transmitter u, then

$$\forall u \in V\colon\; P_{uz} > P_{yz}, \qquad P_{uz} \geq P_{yz}\cdot(\gamma+1)^k, \quad k \in [1, A-1]. \tag{6.11}$$

and for each value of k, there can be just a single interferer. If this is not the case, one or more of the requirements in Eqn. 6.9 are violated.

6.3.3 Defining Which Communications Can Co-Exist

Given the above requirements on the interferers, we define the link pairs that cannot be activated simultaneously (i.e., **block** each other). We say that links $\ell_1 = (u, v)$ and $\ell_2 = (x, y)$ block each other if:

- $P_{xv}{<}P_{uv}$ but $P_{xv}{\geq}P_{uv}/\gamma$: the weak interferer causes more interference than the weak interference budget.

- $P_{xv}{>}P_{uv}$ but if $P_{xv}{\leq}\{P_{uv} + N\}{\cdot}\gamma$: strong interferer cannot be removed. ($k{=}0$ in Eqn. 6.11)

- $x{=}u$: these links have the same transmitter.

- $x{=}v$ or $y{=}u$: a node is the transmitter of one link and the receiver of the other.

Note here that, if a link ℓ_1 is activated, a link that blocks ℓ_1 cannot be activated simultaneously. However, this does not imply that a link (say ℓ_j) that does not block ℓ_1 can be activated with probability 1; it depends on (a) whether there are other links of similar interference power as ℓ_j has already been activated at that time and (b) whether its weak interference budget is satisfied.

6.4 Summary

In this chapter we determined the conditions that render SIC efficient; these are related to the maximum number and the strength of simultaneously interfering transmissions. In the next chapter we design topology control algorithms that satisfy the conditions that are identified in this chapter.

Chapter 7

The MU-SIC Framework

In this chapter, we propose centralized and distributed solutions to the MU-SIC problem that was stated in the previous chapter. The common approach with both proposals is to divide the communications in the multi-hop network into sub-topologies, such that any transmission in a given sub-topology is successfully received at the target receiver, with a high probability. The successful reception is provided by limiting the strong and weak interference that is projected on every link.

If each formed sub-topology consists of a few links, many sub-topologies are needed to include all communications and as a result, the delay between the transmission opportunities on a link increases. MUSIC ensures that the number of selected sub-topologies is kept small, while enabling efficient SIC at the receivers. With MUSIC, the number of sub-topologies formed is at most 17% larger than the optimum number of topologies, discovered through exhaustive search (in small networks). MUSIC also outperforms approaches that simply consider the number of antennas as a measure for determining the links that can be simultaneously active.

7.1 Introduction

MU-SIC problem (defined in Chapter 6) seeks to find a minimal number of groups of communications, such that in every group, even if all communications are active, the interference projected by the transmitters on the receivers is bounded. A solution to MUSIC must satisfy:

- The simultaneous activation of the links in each sub-topology constructs signals with sufficiently disparate reception powers. Hence, the receivers are able to filter out stronger signals (perhaps unwanted) and detect the desired data stream(s) using SIC.
- The links that cannot successfully co-exist are included in different topologies they are activated at different times.
- The union of these topologies spans all the links in the multihop network.

In this chapter we introduce the MUSIC framework, which finds the sub-topologies of the multi-hop network, such that in a given sub-topology each communication is successful with a high probability. A communication is successful if there is a bounded number of strong interferers, each of which can be removed using SIC, and the residual interferers (the weak interferers) are small enough such that the target transmitter satisfies a certain SINR threshold.

Specifically, given a link Chloe→Jack and a set of neighbor links, MUSIC successively identifies and adds into the same sub-topology: (a) the set of weaker links that can be active concurrently with Chloe→Jack while ensuring that Chloe's signal is successfully received by Jack with high probability; and (b) the set of stronger links that can be active concurrently with Chloe→Jack can be iteratively removed from the composite signal in order for Jack to extract Chloe's signal.

We show that our centralized topology control approach C-MUSICdeviates from the optimal approach by a factor of $O(\Omega)$ in the worst case; Ω is a function of the number of maximum number of links that interfere with a given link (we provide an exact definition later). We also propose a distributed variant of MUSIC and call it D-MUSIC. With D-MUSIC, nodes exchange information with regards to the strong and weak interference they can accommodate (which is referred to as the strong and weak interference budgets, respectively). With the local knowledge gathered from one-hop neighbors, node pairs identify their communications with different sub-topologies. In Section 7.4 we examine the performance of MUSIC via simulations across multiple topologies and with different numbers of antenna elements. These simulations demonstrate that MUSIC forms sub-topologies that enable efficient SIC operations. We also show that MUSIC outperforms approaches that simply consider the number of antenna elements as a determining factor for allowing parallel transmissions.

We perform extensive OPNET simulations to evaluate both C-MUSIC and D-MUSIC. Our simulations demonstrate that with small-scale topologies, the design of MUSIC is efficient in terms of constructing limited populations of sub-topologies. Specifically, we observe that with C-MUSIC, the number of sub-topologies created is at most 17% larger than those constructed through an exhaustive search (optimal) strategy. With D-MUSIC, this is number is at most 23% higher than the optimal. We also show that D-MUSIC scales well, in terms of bounding the SIC decoding error probability, in a large set of different topologies, with SINR thresholds and varying numbers of antenna elements. We also compare MUSIC against an approach which is agnostic to the SINR requirements and activates $A - 1$ interfering flows in every contention region. We show that MUSIC outperforms such an approach by offering up to 4 times increase in the achievable aggregated network throughput.

7.2 C-MUSIC: Centralized Solution to MU-SIC Problem

7.2.1 Algorithm Description

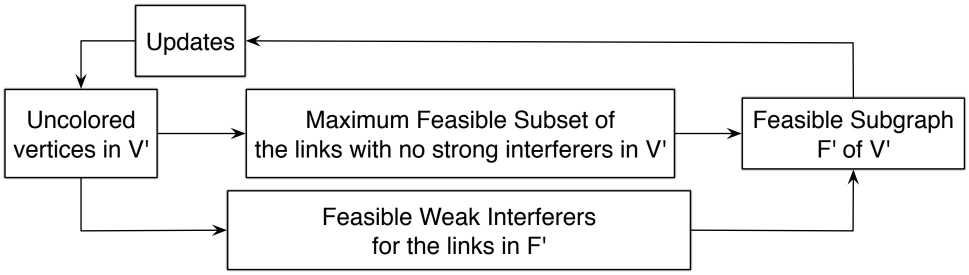

Updates
Uncolored vertices in V'
Maximum Feasible Subset of the links with no strong interferers in V'
Feasible Subgraph F' of V'
Feasible Weak Interferers for the links in F'

Figure 7.1: An outline of the flow of functions towards constructing the feasible sub-topologies with C-MUSIC.

In 6.2.2 we have shown that the problem of finding the minimum number of sub-topologies, each of which can bound the probability of SIC decoding error, is NP-hard. In order to solve this problem, we propose a centralized approximate approach that we call Centralized MUSIC (or C-MUSIC). C-MUSIC takes as input the interference graph $G'(V', E')$. It maintains a dynamically changing set of nodes R' which, to begin with, contains the nodes in G' (i.e., V'). In iterations, it identifies the maximal subset of R' as a valid sub-topology, and removes this subset from consideration in the subsequent iterations. C-MUSIC terminates when all nodes in V' have been associated with sub-topologies. This process is outlined in Figure 7.1.

The main idea behind C-MUSIC is to form a sub-topology R'_i of R', by (i) maintaining a set (say S) of links that do not have strong interferers, (ii) ensuring that the links in S can co-exist successfully, (iii) excluding from R', all the links that cannot co-exist successfully with S so that they are not considered for addition to R'_i, and (iv) including as many links as possible from R', that can *remove* the strong interference from the links in S, and that the links in S can accommodate the weak interference projected by these links. In the following, we first describe how the links in the first sub-topology are chosen. We then elaborate on how the same procedure is used with minor modifications for the latter iterations.

Algorithm 1: C-MUSIC

```
Input: Interference graph $G' = (V', E')$, Error Prob $\delta$
Output: $m$, Number of sub-topologies
begin
    init: $R' = V'$,  $i = 1$;
    repeat
        $R'_i \leftarrow$ Find Sub-Topology($R'$);
        $R' \leftarrow R' - R'_i$; $i \leftarrow i + 1$;
    until $R' = \emptyset$ ;
    return $i$
end
```

Step 1: Adding to a sub-topology, the communications that do not require the removal of strong interferers: C-MUSIC identifies a set S of vertices from R', whose (vertex) weights are greater than all the incoming edge weights. The elements of S represent the communication links whose receivers do not need to remove the strong interferers to decode their target signals. C-MUSIC then finds a maximally feasible subset[1] of S; this subset is called S'. S' is added to set R'_1 of the links corresponding to these vertices are included in the first sub-topology; R'_1 is initially empty. Elements of $\{S - S'\}$ will no longer be considered for being included

[1] Note that a maximal feasible subset is not the same as the *maximum* subset that satisfies the desired properties (using Algorithm 3); finding the maximum feasible subset (*i.e.,* a maximal subset of the highest cardinality) is NP-hard [103].

in R'_1. Furthermore, links that are blocked by the elements of S' ($B(S')$) are also removed from consideration. This step is represented in lines $2-6$ of Algorithm 2.

```
Algorithm 2: Finding sub-topology $R'_i$ of $R'$
   Input: $R'$, (copy of) $G' - (R'_1 \cup R'_2 \cup \ldots \cup R'_{j-1})$
   Output: $R'_i$,  $i^{th}$ sub-topology
   begin
1      init: $R'_i = \emptyset$
2      $S \leftarrow$ {Links in $R'$ that do not have strong interferers in $R'$}
3      $R' \leftarrow R' \setminus S$
4      $S' \leftarrow$MaxFeasibleSubset$(S, \delta)$
5      $R'_i \leftarrow R'_i \cup S'$
6      Refine $(R' - R'_i)$ based on $R'_i$ // update weak interference budgets
7      while $R'$ is not empty do
8          $S \leftarrow$ {links in $R'$, which do not have strong interferers in $(R'$-$R'_i)$}
9          $R' \leftarrow R' \setminus S$
10         $S' \leftarrow$ MaxFeasibleSubset(S);
11         $R'_i \leftarrow R'_i \cup$ FeasibleWeakInterferers$(R'_i, S', \delta)$
12         Two-Way-Updates$(R', S', R'_i)$
       end
13     return $R'_i$
   end
```

Step 2: Updating weak interference budgets: The elements of R'_1 are traversed. For each considered element v, if any element (say u) in $R' - R'_1$ projects weak interference on v, the weak interference budget of v is decremented by $w_{E'}(u, v)$. This step decreases the "maximum additional weak interference" that can be tolerated on link v.

Step 3: Finding the candidate vertices for the next round of SIC: From the updated set R', C-MUSIC finds a set of vertices that can perform SIC to eliminate strong interference from the current set R'_1. The elements in R' are considered sequentially. For each element $e \in R'$, if there is an element $v \in R'_1$ such that $w_{E'}(v,e) > w_{V'}(e)$, v is a strong interferer; the maximum k is found, such that $w_{E'}(v,e)/w_{V'}(e) > (1+\gamma)^k$. If either of the following conditions do not hold for e, it is removed from the set R': (i) There is no other element in R'_1, that yields the same value of k (found above) for removing the strong interference on e. (ii) k is not zero. This step is carried out by instructions 8 and 9 in Algorithm 2.

Algorithm 3: Algorithm MaxFeasibleSubset

```
Input: Set of Links $S$ that do not need SIC, Error Margin $\delta$
Output: $S'$, subset of $S$ in which each link's SINR is satisfied w.p. 1-$\delta$
begin
    init: $S' = \emptyset$
    repeat
        $\ell \leftarrow$ the most-blocking link in $S$
        $S \leftarrow S \setminus \{\ell\}$
        $S' \leftarrow S' \cup \{\ell\}$;        $r_{V'}(\ell) = w_{V'}(\ell)$
        $S \leftarrow S \setminus \{\psi \in S$:  $\ell$ blocks $\psi$  (or $\psi$ blocks $\ell$)$\}$
        foreach  $z \in S$  s.t.  $(\ell, z) \in E'$  do
            $r_{V'}(z)$ -= $w_{E'}(\ell,z)$    // residual interference of $z$
            $S \leftarrow S \setminus \{v\}$ if $r_{V'}(z) < 0$
        end
    until $S = \emptyset$ ;
    return $S'$
end
```

Step 4: Populating the sub-topology with as many elements as possible that can perform SIC: The elements that remain in R' at the end of Step 3 can co-exist with the elements in R'_1. Because Step 3 accounts for the SIC of the strong interferers in R'_1, the strong interference caused by R'_1 on residual R' is assumed to be zero. This assumption facilitates the construction of a subset of R' with a similar procedure as in Step 1; the set of links which do not have strong interferers in $R' \cup R'_1$ are found. From these vertices, C-MUSIC again finds a set S where SIC is not needed. It then finds the maximal feasible subset S' of S using Algorithm 3 as in Step 1. Subsequently, the elements in R'_1 are examined in terms of whether their weak interference budget can accommodate all elements in S'. In particular, we examine whether the condition in Eqn. 6.8 is satisfied for *each* element v in R'_1. For this, we find the maximum feasible weak interference $w_{max_{E'}}$ such that if all weak interferers in S' projected an interference of $w_{max_{E'}}$ on v, v could accommodate all of these interferers (Line 5 of Algorithm 4). $w_{max_{E'}}$ is found, as we check whether Eqn. 6.8 is satisfied for *all* the elements in S' that cause weak interference on v. As long as Eqn. 6.8 is not satisfied, the element in S' that causes the maximum weak interference is removed from S'; we continue to check with the remaining S' until the equation is satisfied. At the end of this process, all remaining links in S' are included in R'_1. Step 4 is captured by Algorithm 4, and lines 11-12 in Algorithm 2. The procedure *Two_Way_Updates()* is used to find the set of candidates for the next round and update the set R'_1 to include the set of links that were successfully included in this round.

Algorithm 4: Algorithm FeasibleWeakInterferers

```
Input: Sub-topology $R'_i$, Candidate Links $S'$, Link Error Margin $\delta$
Output: $F'$, the subset of $S'$ that can be added to $R'_i$
begin
    init: $F' = S'$;
    while $S'$ is not empty do
        $\ell \leftarrow \{z \in R''_i$ that blocks most links in $S'\}$
        $n_\ell = |\psi|$,    where  $\psi \in S' : w_{E'}(\psi,\ell) < w_{V'}(\ell)$
        $(n'_\ell, w_{max_{E'}}(\ell, z)) \leftarrow$ErlangFeasible()$(n_\ell, r_{V'}(\ell))$;    $w_{max_{E'}} = w_{max_{E'}}(\ell, z)$;
        $C \leftarrow C \setminus \{v \in C : w_{E'}(v, \ell) > w_{max_{E'}}\}$
        foreach $z \in S'$ do update residual capacity (i.e. remaining affordable interference)
        of $z$ now that $\ell \in S'$
    end
    return $S'$
end
```

Continuing with subsequent rounds: The steps above are then repeated and in each round, a subset of vertices from R' are added to R'_1 and a new subset is eliminated from contention. The process terminates if R' is empty at this time; *i.e.,* each link is either included in sub-topology 1 or is eliminated from consideration for R'_1. If not, the process returns to Step 3 above.

Constructing subsequent sub-topologies: Once the first sub-topology R'_1 is determined, R' is updated to remove these vertices and the links incident on them. Residual R' contains those links that could not be included in the first sub-topology. These links are considered for the second sub-topology and the above steps are repeated. At the end of the process, R'_2 is determined. If there are links in R'_1 that can be activated in addition, in the second sub-topology (they do not violate the requirements of the R'_2 links that are already included), they are included as well using a procedure similar to the above (not described in detail due to space constraints). Similarly, when links are chosen for sub-topology j (R'_j) links that are in $R1' \cup R2' \ldots R'_{j-1}$ are considered for reactivation in sub-topology j. In other words, after fulfilling the fairness constraints the algorithm tries to *pack* as many links as possible into each sub-topology.

7.2.2 Performance Bound of C-MUSIC

Next, we show that our approach has a *performance efficiency* that is within Ω of the optimal, where Ω is the *maximum opportunity cost* in the network. We first define a few key terms (including performance efficiency and opportunity cost) and later, prove the performance bound.

Interfering Links: Eqn. 6.3 determines whether two distinct links can be active simultaneously without SIC. If the SINRs measured at the two receivers are simultaneously higher than γ, the two links can be active together; if not, the two links are classified as *interfering* links. Note that *interfering links* have an edge between them in the interference graph. Let $\mathbf{T}_l$ denote the set of interfering links for link l. If link l is active, none of the links in $\mathbf{T}_l$ can be active.

The opportunity cost: The *opportunity cost* O_l for link l is the maximum number of links in $\mathbf{T}_l$ that can be active simultaneously if l is deactivated. From among the opportunity costs of all the links in the network, we denote the maximum opportunity cost[2] by Ω.

Definition 7.2.1 *We define the* performance efficiency *of an algorithm as the ratio of number of links selected at the end of the algorithm to the total number of candidate links at the beginning of the algorithm.*

[2] In [17], for the SISO case, Ω is shown to follow $O(|V|^{1-\frac{2}{\varphi(\theta)+\epsilon}}(\log|V|)^{\frac{2}{\varphi(\theta)+\epsilon}})$, where θ is the path loss exponent, $\varphi(\theta)$ is a constant which depends on θ, and ϵ is an arbitrarily small positive constant.

We first consider the phase where, the set of nodes that cannot apply SIC are chosen. At the end of this phase, a sub-set of such nodes (that cannot perform SIC given this chosen set) are *blocked* i.e., removed from consideration. In addition, the SIC condition is applied to choose a set of candidate nodes, that are considered for the subsequent phase. Let us call this Phase 1. Let α' and α^* be the performance efficiencies of the algorithm that performs Phase 1 (consisting of the MFS selection and the selection of nodes for the next phase) and and the optimal algorithm for Phase 1, respectively. Then, the following lemma is holds.

Lemma 7.2.2 *The algorithm that performs Phase 1 achieves a performance efficiency* [3] *of* $1/\Omega$*. In other words,* $\alpha^* \leq \Omega \cdot \alpha'$ *is satisfied.*

Proof 7.2.3 *Consider link* l*, the link that blocks the most links; this is the link that is chosen first by Algorithm 3 when finding a maximal feasible subset. If this link is added to the sub-topology, then no link that is blocked by this link from among the subset of nodes that are considered by Algorithm 3 can be active simultaneously. Furthermore, the activation of this link (perhaps jointly with other links in this subset) can block links not in this subset, from performing SIC. If link* l *were to be deactivated, at most* Ω *additional links can be active in lieu of link* l*, where* Ω *is defined as above; note that* Ω *being an upper bound subsumes the possibility of SIC on links that have an edge to link* l *in the interference graph. Applying the same argument inductively, the proof is established.*

The following theorem provides a performance bound on our proposed algorithm.

Theorem 1 *Algorithm 2 has a performance efficiency of* Ω*.*

[3] This is the maximum ratio by which the results of an approximation algorithm may differ from the optimal solution.

Proof 7.2.4 *For each sub-topology, the proposed algorithm (Algorithm 2) executes in two phases. Phase 1 was define above. In Phase 2, the algorithm checks if, for the set of candidate nodes, the weak interference budgets of nodes already selected are violated. The set of procedures in Phase 2 are similar; an MFS is selected from the candidate set, and nodes that block the already selected nodes are removed. Leveraging Lemma 7.2.2, it is easy to see that the performance efficiency of Phase 2 and the optimal algorithm are simply α' and α^*, respectively.*

Let S^* and C^* (or S' and C') denote the outputs achieved at the end of Phase 1 and Phase 2 by the optimal algorithm (or by Algorithm 2), respectively. Then, in order to prove the theorem, we need to show that:

$$\frac{|S^*| + |C^*|}{\Omega} \leq |S'| + |C'|. \tag{7.1}$$

We denote R^* and R' to be the number of candidate links for a sub-topology at the beginning of the optimal algorithm and Algorithm 2, respectively. Since the optimal algorithm finds at least as many links as the proposed algorithm for a sub-topology, the set of candidate links for forming the subsequent sub-topology, is smaller. In other words, $|R^*| \leq |R'|$. With this:

$$\begin{aligned} \frac{|S^*| + |C^*|}{|S'| + |C'|} &= \frac{\alpha^*|R^*| + \alpha^*(|R^*| - \alpha^*|R^*|)}{\alpha'|R'| + \alpha'(|R'| - \alpha'|R'|)} \\ &= \frac{|R^*|}{|R'|} \cdot \frac{2\alpha^* - (\alpha^*)^2}{2\alpha' - (\alpha')^2} \leq \frac{2\alpha^* - (\alpha^*)^2}{2\alpha' - (\alpha')^2}. \end{aligned} \tag{7.2}$$

Using Lemma 7.2.2 reduces the last term in Eqn. 7.2 to

$$\frac{2\alpha^* - (\alpha^*)^2}{2\alpha' - (\alpha')^2} = \frac{\alpha^*}{\alpha'} \cdot \frac{2 - \alpha^*}{2 - \alpha'} \leq \Omega$$

7.3 D-MUSIC: Distributed Solution to MU-SIC Problem

The centralized approach of C-MUSIC requires information collection and their evaluation at a central controller, towards forming the feasible sub-topologies. In uncontrolled multi-hop networks without an infrastructure, it is preferable that each node processes local information to give decisions and exchange decisions with their neighbors to enhance their local knowledge; consequently, nodes' distributed decisions form sub-topologies that are similar to the ones formed by the central controller. To address this need, we propose D-MUSIC, a distributed approach for constructing the feasible sub-topologies. While D-MUSIC does leverage some of the features of C-MUSIC, there are modifications to enable distributed operations. The key property of D-MUSIC is that each node determines the links that can be activated concurrently in their neighborhoods based on the information they gather via exchange of local messages. Thus, while the sub-topologies are determined on a global scale, the process of determination is performed in a distributed manner, in parallel, and locally in the different parts of the network.

Assumptions: We assume that nodes are aware of their one-hop neighborhoods. In particular, using a neighbor discovery process of some sort (such as HELLO messages [70]), we assume that a node knows who its neighbors are and the signal power on the link to each of these neighbors. A potential transmitter uses the received feedback (to its pilot tones) from its neighbors for selecting the best antenna for communicating with these neighbors. For ease of presentation, we assume that every node has packets to send to each of its neighbors. The approach can be easily modified to accommodate the case where only a sub-set of the nodes have packets to send (or the case when each node has packets for a subset of its neighbors).

We assume that the message exchange during D-MUSIC is performed via SISO transmissions. Multi-user MIMO communications with selection diversity and SIC are activated during a data packet exchange phase that follows the sub-topology formation phase. In the data exchange phase, we allow nodes to transmit packets at random, using an ALOHA-like protocol. We discuss about the indifference of the success of D-MUSIC to the MAC protocol in use, in Section 7.5.

A feasible sub-topology is formed, as each node (i) designates itself to become a transmitter, or (ii) accepts to receive from a designated transmitter, or (iii) determines itself to be idle (because neither a successful transmission nor a successful reception is possible in the formed sub-topology). As with C-MUSIC, the strong and weak interference budgets of the individual nodes determine these decisions. In each subsequent sub-topology, the communications (*i.e.,* transmitter-receiver pairs) that were not previously associated with a sub-topology are considered **with a higher priority**. After including as many of these communications as possible to the sub-topology under construction, the communications that are already associated with other sub-topologies are **re-considered**. This approach increases the spatial reuse. Furthermore, it increases the frequency of link activity especially in sparse regions of the network; in these regions, all communications are associated with .

Transmitter functions: If a node u is not a priori chosen to be a receiver in a given sub-topology, it can be a transmitter, unless one of these conditions is violated:

(i) the transmission of u must not exceed the weak interference budget of any of the receiver nodes in the same sub-topology, and (ii) the transmission of u must not cause a strong interference at any neighbor receiver (say z), causing the SIC-based decoding to fail at this receiver. The latter implies that either this receiver cannot decode the strong interference from u (*i.e.,* k=0 in Eqn. 6.11), or that another transmitter in the same sub-topology causes a strong interference, which is decoded in the same SIC iteration as v (the same value of k in Eqn. 6.11). If any of these conditions is violated, u is precluded from initiating the transmission process.

If transmitter u *can* become a transmitter, it chooses one of its neighbors as a candidate receiver. Among the *possible* nodes (*i.e.,* that are not already a transmitter in the same sub-topology or have not announced themselves to be reception-disabled), and based on the priority criteria described earlier (communications that are not associated with an earlier sub-topology are considered before re-considering others), transmitter node chooses its best (in terms of signal strength) neighbor v as a candidate receiver.

After a randomly chosen time period, u transmits a packet to v; this packet includes the IDs of u and v, and the sub-topology in which u wishes to communicate with v. The transmitter functionalities are depicted in Figure 7.2.

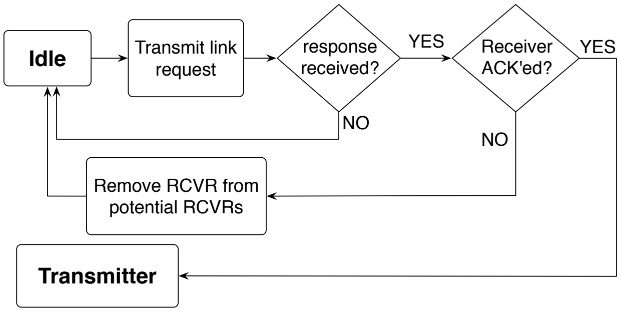


Figure 7.2: The operations of the Transmitter nodes with D-MUSIC.

If the transmitter u does not hear a response message from v corresponding to the current sub-topology for a pre-defined time period, it may try to reactivate some of its links which were included in a prior sub-topology. The action is based on a implicit understanding that there are no additional links being activated in the current sub-topology in the transmitter's neighborhood; clearly, the weak interference and strong interference constraints (as discussed above) need to be satisfied in order for a transmission to be initiated.

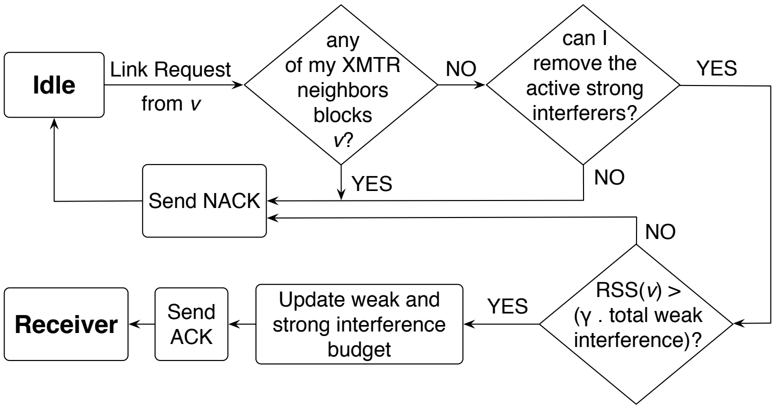

Idle
Link Request from v
any of my XMTR neighbors blocks v?
NO
can I remove the active strong interferers?
YES
YES
NO
Send NACK
NO
RSS(v) > (γ . total weak interference)?
YES
Update weak and strong interference budget
Send ACK
Receiver

Figure 7.3: The operations of the Receiver nodes with D-MUSIC.

Receiver functions: Upon the receipt of the above packet, the receiver v determines if the proposed transmission is viable. Towards this, v checks whether this packet can be decoded in the indicated sub-topology given the local knowledge of which communications are already included in that sub-topology. If the reception necessitates the SIC-based removal of any strong interferers, v determines if this removal is feasible given the transmitters in its neighborhood. Note that the transmitter may not be aware of all the strong interferers in the receiver's neighborhood. If v deems the reception on link (u, v) is feasible in terms of both the weak interference and the strong interference terms that are projected on this communication, it transmits an acknowledgement to u, confirming the addition of link (u, v) to the proposed sub-topology. The acknowledgement message contains the residual weak and strong interference budgets of v after adding link (u, v). The latter is the remaining available values of k (Eqn. 6.11), *i.e.,* the SIC iterations that are not used for removing strong interferers. We show the receiver functions in Figure 7.3.

If v deems the reception on link (u, v) is infeasible, it responds with a negative acknowledgement message. The reception of a negative-ACK informs u that v cannot decode its signal. If node u does not receive either positive or negative acknowledgement from v, it will repeat its request for adding the link (u, v) one more time.

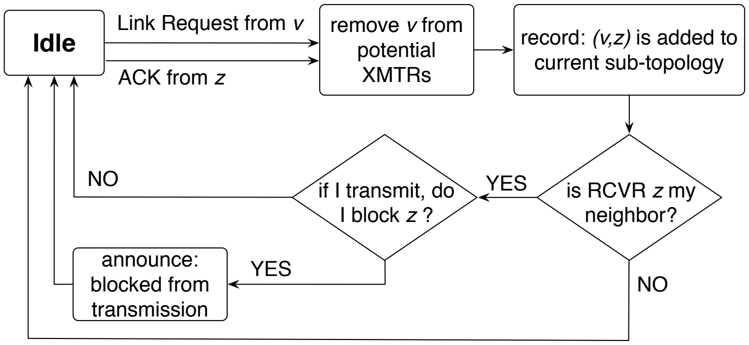

Idle
Link Request from *v*
ACK from *z*
remove *v* from potential XMTRs
record: *(v,z)* is added to current sub-topology
is RCVR *z* my neighbor?
YES
if I transmit, do I block *z* ?
NO
YES
announce: blocked from transmission
NO

Figure 7.4: The operations of the Overhearer nodes with D-MUSIC.

Overhearing nodes: Nodes that are neighbors of a node attempting to become a transmitter in a given sub-topology take a conservative approach and assume that the communication will take place in that sub-topology. As a result, they do not consider this transmitter as a candidate receiver in this sub-topology, even if this node does not become one, due to a negative acknowledgement from its target receiver. Nodes that are neighbors of the receiver hear the ACK message. After hearing (from the transmitter or the receiver) a new link is added, all nodes determine whether or not they can initiate a new transmission in the announced sub-topology, given their updated local information for this sub-topology. If they cannot initiate new transmissions (due to this newly activated link), they are precluded from being transmitters for that sub-topology; they can only initiate new transmissions in the next sub-topology. We show these functions in Figure 7.4.

Termination of the process: After a node is able to activate all of its outgoing links in different sub-topologies, it transmits a message announcing that it is ready for the data transfer phase. From the perspective of a node, the data transfer phase does not begin until all of its neighbors have announced that they are ready as well. Inductively, this phase begins only when all the nodes in the network are ready.

7.4 Performance Evaluation

We evaluate the performance of C-MUSIC and D-MUSIC on both specific and random topologies via extensive simulations. In small topologies, we compare the performance of C-MUSIC and D-MUSIC with that of the optimal activation using an exhaustive search algorithm. We also compare the performance of our approaches with an approach where, A-1 other links are allowed to be simultaneously active in the neighborhood of a receiver; as discussed earlier, many MAC protocols are based on allowing as many concurrently active links as the number of antennas on the nodes [87, 65]. We call this approach as the degree of freedom (DoF)-based approach, where the degrees of freedom are the number of antenna elements on the nodes.

7.4.1 Simulation Setup

We implement all the algorithms on the OPNET v.14 simulator; we provide the details below.

Phases of operation: As described earlier, MUSIC is a framework that consists of two phases. Thus, to provide a basis for comparison with the other solutions, we simulate all schemes in two phases. First is a topology control phase where the specific, considered approach (either C-MUSIC, D-MUSIC, Exhaustive Search or DoF-based topology control) is invoked. At the end of this phase, the sub-topologies that are formed are activated sequentially in time for data transfer, which constitutes the second phase. In this phase, the data packets are transmitted between randomly chosen source-destination pairs that are active within the corresponding period of each sub-topology.

Channel Models: We assume that packets experience path loss with a path loss exponent of 4. During the topology control phase, we assume that the MIMO diversity mode [14] is used to increase the reliability of packet transmissions; due to the increased reliability, we assume that the transmissions are lossless. Lossy transmissions due to fading (if SISO were used) would affect all the topology control algorithms in terms of convergence times. Packets in the data transfer phase are subject to both path loss and temporally varying Rayleigh fading.

SIC implementation: When decoding a data packet, we consider all other packets that interfere with the considered packet due to either full or partially overlapped receptions. For a target packet if (a) the strong interference from all such interfering packets can be removed in distinct SIC iterations (have different k values as per Eqn. 6.11) and, (b) the total weak interference from such packets can be accommodated (is below the packet's weak interference budget), the packet is successfully decoded.

Scenarios of Interest: We primarily vary the node density in the network; we use the average node degree as a measure of this density. We consider cases where nodes have 3,4, or 8 antenna elements. We also consider the impact of varying the SINR threshold; we consider two different threshold values, 5 and 12 dB. Nodes generate data packets destined to all of their neighbors. The load generated at a node varies between 10 and 30 packets/sec; each packet is 1500-bytes long. We repeat every simulation experiment for 40 times.

Metrics: We evaluate C-MUSIC and D-MUSIC in terms of the *number of topologies formed* as compared to the optimal (using exhaustive search) in small scenarios. In large scenarios, the exhaustive search takes a long time and does not converge. We also compute the *average decoding success probability*; note that due to temporal variations in channel conditions there will be decoding failures. This could be both due to a signal getting attenuated due to fading or due to an interfering signal having a higher than average power[4]. We also measure the achieved throughput in the data transfer phase with all the considered approaches.

7.4.2 Simulation Results

We evaluate the MUSIC framework, both in terms of its efficiency in constructing the feasible sub-topologies and in terms of the throughput that can be achieved in the network using these sub-topologies. We elaborate on our experiments and our observations in the following.

1. C-MUSIC and D-MUSIC are efficient in terms of the number of sub-topologies formed: We compare the number of sub-topologies formed with C-MUSIC and D-MUSIC to *the minimum* number of topologies found with exhaustive search. Due to the exponential growth with exhaustive search, we only perform our comparisons with small topologies consisting of 40 links (which correspond to either 7 or 8 nodes in different scenarios that we consider). We consider 30 such topologies.

[4] Recall that the sub-topologies were all formed based on mean signal power criteria.

We find that the average number of sub-topologies found using C-MUSIC is 9.18, and the minimum feasible (on average) is 7.83. The average with D-MUSIC is 9.64. These results show that the MUSIC framework can efficiently find sub-topologies; the number found is only marginally higher than that with the optimal.

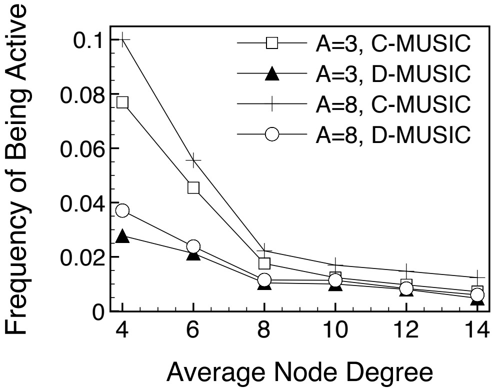


Figure 7.5: The frequency with which the sub-topologies are activated with C-MUSIC and D-MUSIC.

We also compare the frequency with which sub-topologies are activated with D-MUSIC in networks of different densities, with that using C-MUSIC. These results are shown in Fig. 7.5. In small networks, as one might expect, the activation frequencies are high; in larger networks more sub-topologies are formed and the activation frequencies decrease. As the number of antenna elements is increased, SIC is more effective and thus, more links can be grouped under a sub-topology. Consequently, the activation frequency increases with the number of antenna elements used. Most importantly, the performance of D-MUSIC is very similar to that of C-MUSIC.

2. The probability of successful decoding with D-MUSIC does not decrease to a significant extent as density increases: Next we assess the ability of D-MUSIC to provide high decoding success probabilities in scenarios with different levels of interference. For this, we perform simulations with different numbers

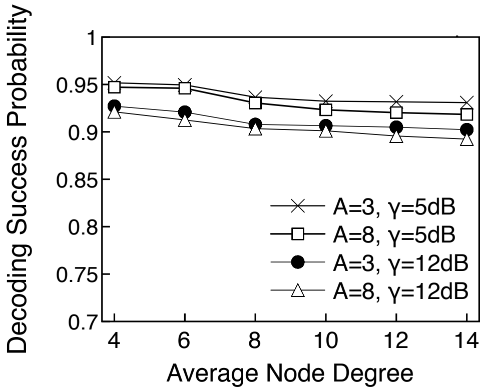


Figure 7.6: The decoding probabilities are stable with C-MUSIC and D-MUSIC for a wide range of densities.

of antenna elements at each node as well as with different SINR thresholds. Fig. 7.6 depicts the results. We observe that with D-MUSIC, the probability of successful decoding remains stable (varies by at most 2%) as the node density increases. This demonstrates that topology control with D-MUSIC in dense multi-user MIMO networks effectively bounds the probability of decoding error.

3. At moderate node densities, a high percentage of the overall throughput can be achieved even before D-MUSIC completely converges: With D-MUSIC, the time taken for complete convergence i.e., until *all* sub-topologies are formed may be long. For high data rate applications, this delay may affect the application performance. We measure the aggregate throughput that is achieved with a *subset* of topologies that are constructed until a specific percentage of the D-MUSIC execution is complete. In other words, we study the convergence of D-MUSIC in terms of activating the links in the topology. We perform the following experiment on a network in which the average node degree is 10; we stop the D-MUSIC after a specific percentage of the total possible sub-topologies are formed. We then proceed to data transfer phase. nodes generate traffic for *all* their neighbors, as they would

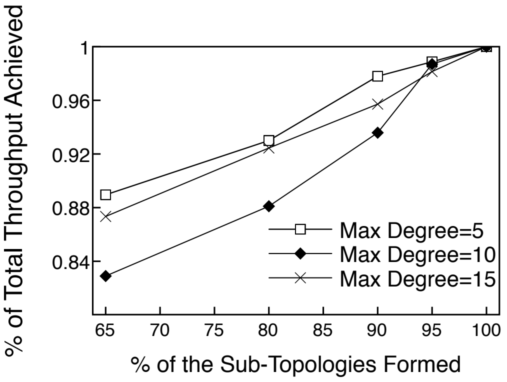


Figure 7.7: D-MU-SIC effectively re-activates links; using only a subset of the sub-topologies, ~80% of the total throughput, which is achieved using all sub-topologies, can be attained.

if D-MUSIC execution was complete. We measure the throughput attained using the subset of links that have been assigned groups in which they become active. We find that D-MUSIC successfully re-activates the activated links, and the throughput achieved with only 80% of the algorithm complete provides almost 90% of the throughput achieved when algorithm teminates (Fig. 7.7).

4. DoF based link activation cannot effectively exploit the benefits of multi-user MIMO: We compare the link activation based on SINR (as in MUSIC) with that simply based on the number of antenna elements (DoF based topology control). For this, we perform simulations on three random topologies of different sizes. We compare the efficiency of C-MUSIC, in terms of successful decoding, with the DoF based technique. DoF forms a sub-topology R_i as follows: (a) A link ℓ is selected as per line 3 of Algorithm 2 and is added into R_i. (b) A new link is added into R_i, as long as either (a) there is no link in R_i that has more than A-1 interfering links, where A is the number of antenna elements on each device, or (b) it does not interfere with any of the links in R_i. The results are depicted in Figure 7.8, Table 7.1 and Figure 7.9. First, we observe that the DoF method allows the concurrent activation of

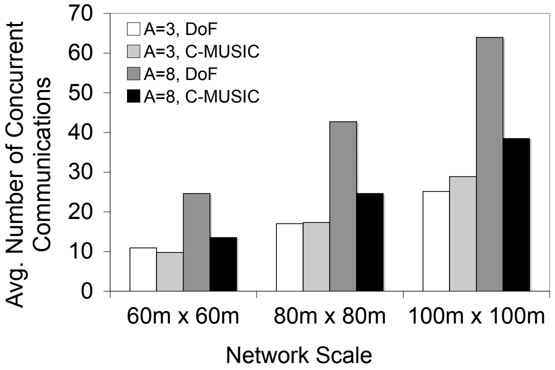


Figure 7.8: The DoF-based link activation method allows more links to be simultaneously active compared to C-MUSIC. When nodes are equipped with many antenna elements, there is a significant gap between the number of links activated using the DoF-based approach and C-MUSIC.

a much larger number of links, as compared to MUSIC (Fig. 7.8), in all the different topologies that we examine. In fact, as the size of the topology increases, DoF allows many more links to be active (as much as 66% more, compared to MUSIC). However, as we observe in Table 7.1, this leads to an excessive increase in the probability of decoding error. In particular, we observe that with 3 antenna elements per node, the application of DoF increases the probability of decoding error by as much as 28 times, while with 8 antennas the increase is 16 times. This expectedly has a direct impact on the total network throughput. In Fig. 7.9 we observe that MUSIC outperforms DoF in terms of total throughput in all considered network densities, by as much as 4 times.

5. The topological properties of the network affect the performance: We perform extensive simulations with both MUSIC and DoF with two types of topologies: a randomly generated topology and a grid. From Fig. 7.10 we see that the efficiency of C-MUSIC is higher with random topologies. With random topologies, fewer sub-topologies are formed, since many more links are amenable to inclusion

Probability of Decoding Error			
		A=3	A=8
3x3	**C-MUSIC**	**0.018**	**0.034**
	DoF-Method	0.616	0.748
4x4	**C-MUSIC**	**0.024**	**0.036**
	DoF-Method	0.643	0.758
5x5	**C-MUSIC**	**0.026**	**0.041**
	DoF-Method	0.680	0.784

Table 7.1: Links activated based on DoF-based topology control experience high decoding errors of up to 68%, whereas the decoding error using MUSIC is bounded with 2.5%. These decoding errors degrade the performance of the DoF-based approach significantly, overriding its benefits shown in Figure 7.8.

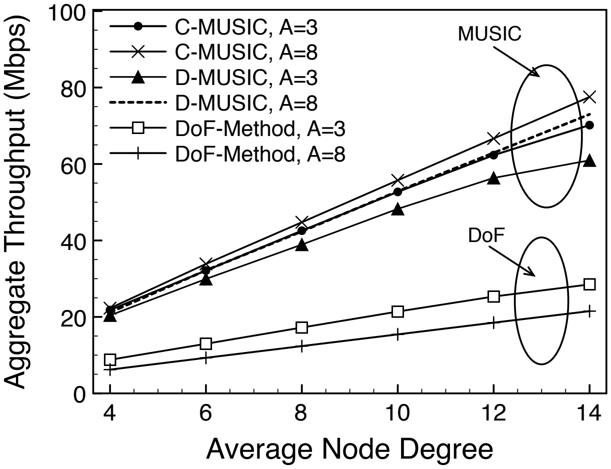


Figure 7.9: MUSIC offers higher aggregate throughput in the multi-hop network, compared to the throughput with the DoF-based approach.

in every sub-topology. On the other hand, in grid topologies, we observe that C-MUSIC is coerced into constructing a significantly larger number of sub-topologies (2.2 times more than with random topologies for A=8), each consisting of much fewer links. This is because in a grid topology each node is equi-distant from its neighbors. Hence, the signals of neighbors arrive at a receiver with similar powers; this increases the probability of decoding error with SIC. Due to this, it is also the case that in the grid deployment, the number of antennas does not affect the number of formed

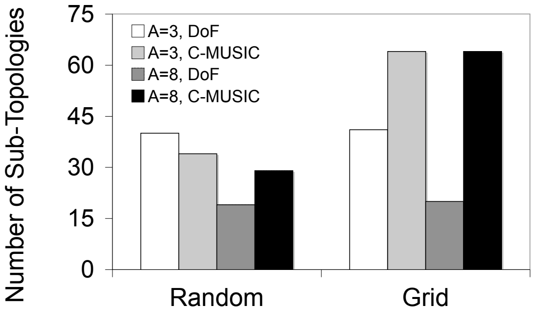


Figure 7.10: The type of topology affects the ability of MUSIC to form limited numbers of sub-topologies.

sub-topologies. Unlike C-MUSIC however, we observe that DoF constructs a similar number of sub-topologies with both the random and the grid deployments. This is because DoF takes into account the node degree (which is 7 on average in all simulated topologies) and not the SINR. However, due to this it suffers from degraded decoding probabilities. We observe that the decoding probability with the DoF-based approach was 0.73, while that with C-MUSIC was 0.013.

7.5 The Scope of MUSIC

In this chapter we proposed MUSIC, a topology control framework for exploiting the benefits of multi-user reception using SIC in MIMO networks. MUSIC has been designed based on the conditions derived in Chapter 6; these conditions represent the physical layer requirements for realizing multi-user MIMO. By observing these conditions, MUSIC is sensitive to the cross-layer interactions between the higher layers and the physical layer.

In the following, we discuss the scope of MUSIC and elaborate on how it differentiates from the prior efforts that study multi-user MIMO communications. As discussed in 2.3.3, most prior efforts for enabling successful multi-user MIMO communications aim to *schedule* all communications in specific time divisions. The topology construction with MUSIC has four major differences compared to a scheduling approach.

MUSIC obviates the need for packet-level synchronization. Packet-level synchronization is an essential component of schedule-based medium access. When communications are scheduled to a particular time, they need to take place within the provided short time frame. On the other hand, it is extremely difficult to synchronize the clocks of a variety of devices in an ad hoc network, which has no infrastructure or centralized control.

MUSIC constructs sub-topologies, each of which are active for longer time frames compared to the duration of communications in the scheduled access. Any link in these sub-topologies may be active once or multiple times while the sub-topology is allowed to be active. Furthermore, any preferred medium access control mechanism can arbitrate the order in which these communications will be active; a simple random access scheme such as ALOHA [6], a carrier sense multiple access mechanism [45], or more specialized medium access control mechanisms can be used. The choice of the most efficient mechanism to control the medium access on the constructed sub-topologies can vary based on the application the multi-hop network is used for. The design of a medium access control protocol is beyond the scope of MUSIC.

MUSIC is more robust to mobility compared to link-scheduling algorithms. The mobility of the environment as well as of the nodes in the network necessitates updates in the constructed schedule or sub-topologies. Node mobility affects both the scheduled medium access of all communications, and the random access of the communications in each sub-topology; however, the impact of environmental mobility on the former is more severe.

For the successful SIC-based decoding of concurrent transmissions at a given receiver, it is important that the receiver has an estimate of the channel gains with respect to all transmitters at the time of transmission. A schedule guarantees correct decoding in each time division; therefore, a scheduler needs to know (apriori) and account for the gains on each channel that will be concurrently used for successful SIC-based decoding in a MU-MIMO system. As the mobility in the environment renders the channel estimates obsolete, a schedule becomes obsolete more quickly compared to the constructed sub-topologies; the latter is robust to the mobility of the environment as we describe next.

With MUSIC, the sub-topologies are constructed using the mean channel gain among node pairs, which is computed in the most recent time-frame. After this construction the sub-topologies become active one by one for data communications; after each sub-topology has become active, starting with the first sub-topology, all sub-topologies become active a second time. These rounds are repeated until there is significant probability of decoding error at the receivers, which indicates that the sub-topologies must be re-calculated with on the updated network topology. This is how MUSIC copes with node mobility.

When a sub-topology is allowed to become active, the receiver nodes of the given sub-topology need to have channel estimates for only the transmitter nodes in this sub-topology. Aside from reducing the complexity of channel estimation process, MUSIC is also more robust to a change of environment. While the *schedule* needs to be re-calculated in the mobility of the environment, the *sub-topologies* remains the same (the receivers cope with the change by updating channel estimates.

Topologies formed by MUSIC can sustain various traffic patterns. In order to construct communication schedules in a MU-MIMO system, the traffic load and patterns of the nodes need to be known. Because scheduling tries to activate links using the minimum number of time divisions, the absense of traffic information either yields the constructed schedule inefficient in terms of the number of time divisions (overestimation of the load leads to under-utilization of time divisions), or nodes' traffic requirements are not met efficiently (underestimation of the load may leave some communications unscheduled).

Topology control approach of MUSIC provides a flexible infrastructure in terms of changing traffic patterns. The sub-topologies are constructed assuming *every* link has a non-zero load at all times. Changes in the traffic patterns reduce the probability of decoding error in a given sub-topology. On the other hand, it is also possible to construct fewer sub-topologies if information with regards to the traffic patterns is available. Obtaining traffic load information incurs an additional messaging overhead and obtaining this information accurately is hard. Because of this, many prior efforts studied scheduling and routing jointly [30, 47, 66]. In our work, we have used the former approach of treating every node to have traffic for all of its neighbors.

MUSIC is conducive to distributed implementation. Finally, a distributed construction of transmission schedules in the context of MU-MIMO networks is extremely hard, as nodes need to have an accurate information with regards to the communication channels within at least two-hop neighborhood. In fast changing channels, this may not be provided in spite of frequent message exchange. On the other hand, D-MUSIC provides a distributed solution that relies only on the availability and exhange of local information.

Chapter 8

Conclusions

In this dissertation, we have investigated approaches for utilizing link-level benefits with antenna arrays in multi-hop wireless networks. We have studied three communication paradigms using antenna arrays on the transmitter and the receiver, namely directional beamforming, MIMO spatial diversity, and MIMO spatial multiplexing. Each of these antenna array communications yields significantly better performance compared to the conventional system (*i.e.,* communication using a single antenna) in terms of capacity and/or robustness. A multi-hop network can benefit from these communications only if higher-layer decisions facilitate them at the physical layer, and this is possible only if the communication characteristics are accurately captured at the higher layers. As a consequence, the overall goal of this dissertation has been to emphasize the importance of a **cross-layer design**.

The cross-layer design of LDS, D-LDS and Di-ATC have targeted the efficient use of fully-directional communications in multi-hop networks. The cross-layer design of M-3P, the physical layer model for representing MIMO-STBC communications at the higher layers, has targeted an accurate mapping of space-time coded MIMO communication characteristics to the higher layers. Finally, the cross-layer design of

C-MUSIC and D-MUSIC have targeted a performance benefit in multi-hop networks that use multi-user MIMO communications.

It is our hope that the work presented here makes a strong statement with regards to the importance of a cross-layer approach, when the goal is to observe higher-layer benefits as a consequence of using a novel technology at the physical layer.

Bibliography

[1] IEEE P802.11 - TASK GROUP N. IEEE 802.11n: Standard for Enhancements for Higher Throughput.

[2] Opnet user's documentation. http://www.opnet.com.

[3] Ralink technology: MIMO. http://www.ralinktech.com.

[4] The MAdWiFi driver. http://madwifi.org.

[5] A. Willig A. Kopke and H. Karl. Chaotic maps as parsimonious bit error models of wireless channels. In *IEEE INFOCOM*, 2003.

[6] N. Abramson. The aloha system: another alternative for computer communications. In *Fall Joint Computer Conference (AFIPS)*, 1970.

[7] Y. Akita and K. Sega. Test strategies for 2-by-2 MIMO in 802.11n systems, 2007. Tektronix Inc. Online Article.

[8] S. M. Alamouti. A simple transmit diversity technique for wireless communications. *IEEE JSAC*, 16(8), 1998.

[9] K. Alzoubi, X. Li, Y. Wang, P. Wan, and O. Frieder. Geometric spanners for wireless ad hoc networks. *IEEE TPDS*, May 2003.

[10] J. G. Andrews and T. H. Y. Meng. Performance of multicarrier CDMA with successive interference cancellation in a multipath fading channel. *IEEE Transactions on Communications*, 52(5):811–822, 2004.

[11] S. Arya, G. Das, D. M. Mount, J. S. Salowe, and M. Smid. Euclidean spanners: short, thin, and lanky. In *ACM STOC*, 1995.

[12] S. Arya and M. H. M. Smid. Efficient construction of a bounded degree spanner with low weight. In *ESA*, 1994.

[13] I. Berenguer, X. Wang, and V. Krishnamurthy. Adaptive MIMO antenna selection. In *Asilomar Conference on Signals, Systems, and Computers*, 2003.

[14] E. Biglieri, R. Calderbank, A. Constantinides, A. Goldsmith, A. Paulraj, and H. V. Poor. MIMO wireless communications, 2007.

[15] P. Bose, L. Devroye, W. Evans, and D. Kirkpatrick. On the spanning ratio of gabriel graphs and β-skeletons. *SIAM Journal on Discrete Mathematics*, 2001.

[16] P. Bose, J. Gudmundsson, and M. Smid. Constructing plane spanners of bounded degree and low weight. In *ESA*, 2002.

[17] G. Brar, D. M. Blough, and P. Santi. Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks. In *ACM MOBICOM*, 2006.

[18] M. Burkhart, P. von Rickenbach, R. Wattenhofer, and A. Zollinger. Does topology control reduce interference? In *ACM MobiHoc*, 2004.

[19] G. Calinescu. Computing 2-hop neighborhoods in ad hoc wireless networks. In *ADHOC-NOW*, 2003.

[20] R. Choudhury and N. Vaidya. Deafness: A MAC problem in ad hoc networks when using directional antennas. In *IEEE ICNP*, 2004.

[21] R. Choudhury, X. Yang, N. Vaidya, and R. Ramanathan. Using directional antennas for medium access control in ad hoc networks. In *ACM MOBICOM*, 2002.

[22] S. Chu and X Wang. Opportunistic and cooperative spatial multiplexing in MIMO ad hoc networks. In *ACM MobiHoc*, pages 63–72, 2008.

[23] D. Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In *ACM MobiCom*, 2003.

[24] T. Elbatt. Towards scheduling MIMO links in interference-limited wireless ad hoc networks. In *IEEE MILCOM*, 2007.

[25] D. Eppstein. Spanning trees and spanners, 1996. Technical Report.

[26] B. Bjerke et.al. Packet error probability prediction for system level simulations MIMO-OFDM based 802.11n WLANs. In *IEEE ICC*, 2005.

[27] G. J. Foschini. Layered space-time architecture for wireless communication in a fading environment when using multi- element antennas, 1996. Bell Labs. Technical J. vol. 1, no.2.

[28] J. Gao, L. J. Guibas, J. Hershberger, L. Zhang, and A. Zhu. Geometric spanner for routing in mobile networks. In *ACM MobiHoc*, 2001.

[29] E. Gelal, G. Jakllari, and S.V. Krishnamurthy. Exploiting diversity gain in MIMO equipped ad hoc networks. In *Asilomar Conference on Signals and Systems*, 2006.

[30] T. Girici and A. Ephremides. Joint routing and scheduling metrics for ad hoc wireless networks. In *Asilomar Conference on Signals, Systems and Computers*, 2002.

[31] O. Goussevskaia, Y. A. Oswald, and R. Wattenhofer. Complexity in geometric SINR. In *ACM MobiHoc*, 2007.

[32] G. Daniels H. Suzuki, M. Hedley and J.Yuan. Performance of MIMO-OFDM-BICM on measured indoor channels. In *IEEE VTC*, 2006.

[33] D. Halperin, T. Anderson, and D. Wetherall. Taking the sting out of carrier sense: interference cancellation for wireless LANs. In *ACM MOBICOM*, 2008.

[34] R. Heath, S. Sandhu, and A. Paulraj. Antenna selection for spatial multiplexing systems with linear receivers. In *IEEE Communications Letters*, 2001.

[35] R. W. Heath and D. J. Love. Multimode antenna selection for spatial multiplexing systems with linear receivers. *IEEE Transactions on Signal Processing*, 53:3042–3055, 2005.

[36] Z. Huang, C. Shen, C. Srisathapornphat, and C. Jaikaeo. Topology control for ad hoc networks with directional antennas. In *IEEE ICCCN*, 2002.

[37] Z. Huang and C.-Chung Shen. Topology control with directional power intensity for ad hoc networks. In *IEEE WCNC*, 2004.

[38] Z. Huang and C.-Chung Shen. Multibeam antenna-based topology control with directional power intensity for ad hoc networks. *IEEE Transactions on Mobile Computing*, 5(5), 2006.

[39] H. Jafarkhani. *Space-Time Coding, Theory and Practice.* Cambridge University Press, 2005.

[40] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance. In *ACM MOBICOM*, 2003.

[41] G. Jakllari, S. V. Krishnamurthy, M. Faloutsos, P. Krishnamurthy, and O. Ercetin. A framework for distributed spatio-temporal communications in mobile ad hoc networks. In *IEEE INFOCOM*, 2006.

[42] G. Jakllari, W. Luo, and S. V. Krishnamurthy. An integrated neighbor discovery and MAC protocol for ad hoc networks using directional antennas. *IEEE Transactions on Wireless Communications*, 6(3), 2007.

[43] D. B Johnson and D. A Maltz. Dynamic source routing in ad hoc wireless networks. *Mobile Computing*, 353, 1996.

[44] L. Kleinrock and J. A. Silvester. Optimum transmission radii for packet radio networks or why six is a magic number. In *Proceedings of the IEEE National Telecommunications Conference*, 1978.

[45] L. Kleinrock and F. Tobagi. Packet switching in radio channels: Part I - carrier sense multiple access modes and their throughput delay characteristics. *IEEE Transactions on Computing*, 23(12), 1975.

[46] Y. Ko, V. Shankarkumar, and N. Vaidya. Medium access control protocols using directional antennas in ad hoc networks. In *IEEE Infocom*, 2002.

[47] M. S. Kodialam and T. Nandagopal. Characterizing achievable rates in multi-hop wireless networks: the joint routing and scheduling problem. In *ACM MOBICOM*, 2003.

[48] T. Korakis, G. Jakllari, and L. Tassiulas. A MAC protocol for full exploitation of directional antennas in ad-hoc wireless networks. In *ACM MobiHoc*, 2003.

[49] D. Kotz, C. Newport, R.S. Gray, J. Liu, Y. Yuan, and C. Elliott. Experimental evaluation of wireless simulation assumptions. In *ACM MSWIM*, 2004.

[50] U. Kumar, H. Gupta, and S. R. Das. A topology control approach to using directional antennas in wireless mesh networks. In *IEEE ICC*, 2006.

[51] M. Levorato, S. Tomasin, P. Casari, and M. Zorzi. An approximate approach for layered space-time multiuser detection performance and its application to MIMO ad hoc networks. In *IEEE ICC*, 2006.

[52] E. Li, J. Halpern, P. Bahl, Y. Wang, and R. Wattenhofer. Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks. In *ACM PODC*, 2001.

[53] G. Li, L. Yang, W. Conner, and B. Sadeghi. Opportunities and challenges for mesh networks using directional antennas. In *IEEE WiMesh*, 2005.

[54] N. Li, J. C. Hou, and L. Sha. Design and analysis of an MST-based topology control algorithm. In *IEEE Infocom*, 2003.

[55] X. Li, G. Calinescu, and P. Wan. Distributed construction of a planar spanner and routing for ad hoc wireless networks. In *IEEE Infocom*, 2002.

[56] X. Li, W. Song, and W. Wang. A unified energy-efficient topology for unicast and broadcast. In *ACM MOBICOM*, 2005.

[57] X. Li, P. Wan, and Y. Wang. Power efficient and sparse spanner for wireless ad hoc networks. In *IEEE ICCCN*, 2001.

[58] X. Li, P. Wan, Y. Wang, and O. Frieder. Sparse power efficient topology for wireless networks. In *HICSS*, 2002.

[59] X. Li, Y. Wang, P. Wan, and O. Frieder. Localized low weight graph and its applications in wireless ad hoc networks. In *IEEE INFOCOM*, 2004.

[60] Y. Li and A. M. Safwat. Efficient deafness avoidance in wireless ad hoc and sensor networks with directional antennas. In *ACM PE-WASUN*, 2005.

[61] J. Liu, Y. T. Hou, Y. Shi, H. D. Sherali, and S Kompella. On the capacity of multiuser MIMO networks with interference. *IEEE Trans. on Wireless Communications*, 7(2), 2008.

[62] T. Liu and P. Vishwanath. Opportunistic orthogonal writing on dirty paper. *IEEE Transactions on Information Theory*, 29, 2005.

[63] M. Min, F. Wang, D. Du, and P. M. Pardalos. A reliable virtual backbone scheme in mobile ad-hoc networks. In *IEEE MASS*, 2004.

[64] A. Miu, H. Balakrishnan, and C.E. Koksal. Improving loss resilience with multi-radio diversity in wireless networks. In *ACM MobiCom*, 2005.

[65] B. Mumey, J. Tang, and T. Hahn. Joint stream control and scheduling in multihop wireless networks with MIMO links. In *IEEE ICC*, 2008.

[66] S. Nahle and N. Malouch. Joint routing and scheduling for maximizing fair throughput in WiMAX mesh network. In *IEEE Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2008.

[67] V. Namboodiri, L. Gao, and R. Janaswamy. Power efficient topology control for wireless networks with switched beam directional antennas. In *IEEE MASS*, 2005.

[68] A. Nasipuri, S. Ye, J. You, and R. Hiromoto. A MAC protocol for mobile ad hoc networks using directional antennas. In *IEEE WCNC*, 2000.

[69] B. Hochwald C. Peel and L. Swindlehurst. A vector-perturbation technique for near-capacity multi-antenna multi-user communication - part ii: Perturbation. *IEEE Transactions on Communications*, 53(3), 2005.

[70] C. E. Perkins and E. M. Royer. Ad hoc on-demand distance vector routing. In *Mobile Computing Systems and Applications*, 1999.

[71] R. J Punnoose, P. V. Nikitin, and D. D. Stancil. Efficient simulation of Ricean fading within a packet simulator. In *IEEE VTC*, 2000.

[72] B. Raman and K. Chebrolu. Design and evaluation of a new MAC protocol for long-distance 802.11 mesh networks. In *ACM MOBICOM*, 2005.

[73] R. Ramanathan. On the performance of ad hoc networks with beamforming antennas. In *ACM MobiHoc*, 2001.

[74] R. Ramanathan, J. Redi, C. Santivanez, D. Wiggins, and S. Polit. Ad hoc networking with directional antennas: A complete system solution. *IEEE Journal on Selected Areas in Communications*, March 2005.

[75] R. Ramanathan, M. Takai, and N. Vaidya. Directional antenna systems for ad hoc networks. Tutorial, *ACM MobiHoc, 2003.*

[76] C. Ribeiro, K. Hugl, M. Lampinen, and M. Kuusela. Performance of linear multi-user MIMO precoding in LTE system. In *IEEE ISWPC*, 2008.

[77] K. Romer. Time synchronization in ad hoc networks. In *ACM MobiHoc*, 2001.

[78] S. Roy, Y. C. Hu, D. Peroulis, and X-Y. Li. Minimum-energy broadcast using practical directional antennas in all-wireless networks. In *IEEE Infocom*, 2006.

[79] S. Roy, D. Saha, S. Bandyopadhyay, T. Ueda, and S. Tanaka. A network-aware MAC and routing protocol for effective load balancing in ad hoc wireless networks with directional antenna. In *ACM MobiHoc*, 2003.

[80] J. S. Salowe. Euclidean spanner graphs with degree four. *Discrete Applied Mathematics*, 54(1), 1994.

[81] N. Santhapuri, J. Manweiler, S. Sen, R. Roy Choudhury, S. Nelakuditi, and K. Munagala. Message in message (MiM): A case for reordering transmissions in wireless networks. In *ACM Hotnets*, 2008.

[82] R. Santosa, B-S. Lee, C. Yeo, and T. Lim. Distributed neighbor discovery in ad hoc networks using directional antennas. In *IEEE CIT*, 2006.

[83] H. Shi, T. Abe, and H. Suda. Iterative power allocation scheme for mimo systems. *IEEE Transactions on Communications*, E-89B:791–800, 2006.

[84] V. Shrivastava, S. Rayanchu, J. Yoon, and S. Banerjee. 802.11n under the microscope. In *IMC*, 2008.

[85] W. Song, Y. Wang, and X. Li. Localized algorithms for energy efficient topology in wireless ad hoc networks. In *ACM MobiHoc*, 2004.

[86] K. Sundaresan and R. Sivakumar. Routing in ad-hoc networks with MIMO links. In *IEEE ICNP*, 2005.

[87] K. Sundaresan, R. Sivakumar, M. A. Ingram, and T.-Y. Chang. A fair medium access control protocol for ad-hoc networks with MIMO links. In *IEEE INFOCOM*, 2004.

[88] R. L. Rivest T. H. Cormen, C. E. Leiserson and C. Stein. Introduction to algorithms, 2001.

[89] M. Takai, R. Bagrodia, K. Tang, and M. Gerla. Efficient wireless network simulations with detailed propagation models. *Wireless Networks*, 7(3), 2001.

[90] M. Takai, J. Martin, and R. Bagrodia. Effects of wireless physical layer modeling in mobile ad hoc networks. In *ACM MobiHoc*, 2001.

[91] M. Takai, J. Martin, R. Bagrodia, and A. Ren. Directional virtual carrier sensing for directional antennas in mobile ad hoc networks. In *ACM MobiHoc*, 2002.

[92] V. Tarokh, H. Jafarkhani, and A. Calderbank. Space-time block codes from orthogonal designs. *IEEE Trans. Info. Theory*, 45(5), 1999.

[93] Alberto Lopez Toledo and Xiaodong Wang. TCP performance over wireless MIMO channels with ARQ and packet combining. *IEEE Transactions on Mobile Computing*, 5(3):208–223, 2006.

[94] D. N. C. Tse, P. Viswanath, and L. Zheng. Diversity-multiplexing tradeoff in multiple-access channels. *IEEE Transactions on Information Theory*, 50(9):1859–1874, 2004.

[95] T. Ueda, S. Tanaka, D. Saha, S. Roy, and S. Bandyopadhyay. A rotational sector-based, receiver-oriented mechanism for location tracking and medium access control in ad hoc networks using directional antenna. In *IFIP PWC*, 2003.

[96] S. Vasudevan, J. Kurose, and D. Towsley. On neighbor discovery in wireless networks with directional antennas. In *IEEE Infocom*, 2005.

[97] S. Verdu. Multiuser detection. In *Cambridge University Press*, 1998.

[98] P. Wan, K. Alzoubi, and O. Frieder. Distributed construction of connected dominating set in wireless ad hoc networks. In *IEEE Infocom*, 2002.

[99] Y. Wang, D.-M. Chiu, and J. C. S. Lui. Characterizing the capacity gain of stream control scheduling in MIMO wireless mesh networks. In *IFIP Networking*, 2007.

[100] Y. Wang and X. Li. Localized construction of bounded degree and planar spanner for wireless ad hoc networks. In *DIALM-POMC*, 2003.

[101] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang. Distributed topology control for power efficient operation in multihop wireless ad hoc networks. In *IEEE Infocom*, 2001.

[102] S. Weber, J. Andrews, X. Yang, and G. Veciana. Transmission capacity of wireless ad hoc networks with successive interference cancellation. *IEEE Transactions on Information Theory*, 53(8), 2007.

[103] D. West. Introduction to graph theory, 1998.

[104] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela. V-BLAST: An architecture for realizing very high data rates over the rich-scattering wireless channel. In *ISSSE*, 1998.

[105] S. Yi, Y. Pei, and S. Kalyanaraman. On the capacity improvement of ad hoc wireless networks using directional antennas. In *ACM MobiHoc*, 2003.

[106] A. Zanella, M. Chiani, and M. Z. Win. MMSE reception and successive interference cancellation for MIMO systems with high spectral efficiency. *IEEE Transactions on Wireless Communications*, 4(3), 2005.