# Fast Distributed Well Connected Dominating Sets for Ad Hoc Networks

Srinivasan Parthasarathy* Rajiv Gandhi†

## Abstract

We present the first distributed algorithms for computing connected dominating sets (CDS) for ad hoc networks that break the linear-time barrier. We present two algorithms which require $O(\Delta \log^2 n)$ and $O(\log^2 n)$ running time respectively, where $\Delta$ is the maximum node degree and $n$ is the size of the network. This is a substantial improvement over existing implementations, all of which require $\Omega(n)$ running time.

A basic primitive which underlies our first algorithm is Distance-2 coloring of the network. This primitive arises naturally in many applications such as broadcast scheduling and channel assignment. Minimizing the number of colors used in the coloring is very desirable for these applications, but this is known to be NP-hard. We present a fast distributed implementation of D2-coloring for ad hoc networks which is guaranteed to use at most O(1) times the number of colors required by the optimal algorithm.

Our algorithms are geared towards constructing Well Connected Dominating Sets (WCDS) which have certain powerful and useful structural properties such as low size, low stretch and low degree. In this work, we also explore the rich connections between WCDS and routing in ad hoc networks. Specifically, we combine the properties of WCDS with other ideas to obtain the following interesting applications:

- An online distributed algorithm for collision-free, low latency, low redundancy and high throughput broadcasting.
- Distributed capacity preserving backbones for unicast routing and scheduling.

Throughout this work, our focus is on obtaining distributed algorithms for ad hoc wireless networks with *provably good analytical performance guarantees*, both in terms of the running times of the algorithms as well as the performance of the broadcast and unicast routing and scheduling applications based on them.

## 1 Introduction

Ad hoc networks are composed of a set of mobile nodes which communicate with one another over a shared wireless channel. Unlike wired networks, nodes in an ad hoc network do not rely on a pre-existing communication infrastructure. Instead, they communicate either directly with each other or with the help of intermediate nodes in the network. The wireless and self-configurable nature of ad hoc networks make them well-suited for several scenarios such as mobile battle-fields, disaster relief, sensing and monitoring. However, the lack of a communication infrastructure introduces several challenging and interesting research issues in the design of communication protocols for these networks.

Several researchers have proposed construction of a *virtual backbone* in ad hoc networks as an analogue of the fixed communication infrastructure in wired networks. A virtual backbone typically consists of a small subset of nodes in the network which gather and maintain information such as local topology and traffic conditions. This information can be made use of by higher level protocols for providing efficient communication services. Connected Dominating Sets (CDS) are the earliest structures proposed as candidates for virtual backbones in ad hoc networks [10, 9, 23]. A dominating set in a graph $G = (V, E)$ is a set of vertices $V' \subseteq V$ such that every node in $V - V'$ is adjacent to some node in $V'$. A CDS is a dominating set whose induced subgraph is connected. An MCDS is a CDS with the minimum number of nodes. In the context of ad hoc networks, a well studied problem is that of finding an MCDS in a Unit Disk Graph (UDG) [26, 7, 3, 4, 2, 5, 1]. UDGs are a class of graphs which are frequently used to model connectivity in ad hoc networks. We use UDGs to model ad hoc networks in this work.

Several distributed CDS algorithms have been proposed for arbitrary undirected graphs and UDGs. These algorithms vary in their running time and message complexity. In general, all existing distributed CDS algorithms can be classified into two categories. The first category of algorithms are fast sub-linear time algorithms such as [11, 27]. However, these algorithms do not consider message losses due to collisions and hence are not directly applicable to ad hoc networks. The second category of algorithms are linear time algorithms. These algorithms can be implemented such that only a single node in the network transmits at any time and hence no collisions occur during the course of the algorithm [26, 3, 4, 2, 5, 1].

A linear time algorithm is undesirable for ad hoc networks due to several reasons. Nodes in a large network may have to wait for long before the CDS can be con-

*Department of Computer Science, University of Maryland, College Park, MD 20742. Research supported by NSF Award CCR-0208005. E-mail: sri@cs.umd.edu.

†Department of Computer Science, Rutgers University, Camden, NJ 08102. E-mail: rajivg@camden.rutgers.edu.

structed. A linear time CDS construction is more susceptible to interruptions due to topology changes which might occur during the course of the algorithm. A linear time distributed algorithm does not exploit the massive parallel processing capability available in the network. In this paper, we propose the first distributed sub-linear time algorithms for constructing a CDS in ad hoc networks. Specifically, we view the following as the main contributions of this work.

### 1.1 Our Contributions

- We present two distributed algorithms for constructing a CDS in ad hoc networks. These algorithms require $O(\Delta \log^2 n)$ and $O(\log^2 n)$ running time respectively, where $\Delta$ is the maximum node degree and $n$ is the size of the network. This is a substantial improvement over existing implementations, all of which require $\Omega(n)$ running time.
- A basic primitive which underlies our first algorithm is the Distance-2 node coloring of the network. This primitive arises naturally in many applications such as broadcast scheduling and channel assignment. In general, the colors could represent time slots or frequencies assigned to the nodes. Minimizing the number of colors used in the coloring is very desirable for these applications, but is known to be NP-hard [22]. As a part of our first CDS algorithm, we present a fast distributed algorithm for D2-coloring ad hoc networks. The running time of this algorithm is $O(\Delta \log^2 n)$ and it uses at most $O(1)$ times the number of colors used by an optimal algorithm.
- The distributed CDS algorithms presented in this paper are geared towards constructing CDSs with certain powerful structural properties such as low size, low stretch and low degree (henceforth, we refer to such a CDS as a Well Connected Dominating Sets (WCDS)). The work by Alzoubi [1] deals with a linear-time distributed construction of WCDS in ad hoc networks. In this paper, we also explore the rich connections between WCDS and routing in ad hoc networks. Specifically, we combine the structural properties of WCDS with other ideas to obtain the following interesting applications:
  - An online distributed algorithm for collision-free, low latency, low redundancy and high throughput broadcasting.
  - Distributed capacity preserving backbones for unicast routing and scheduling.

Our algorithms require the nodes to have knowledge of only local information. Our first algorithm requires that each node know its one hop neighbors, an upper bound on the size of the network $n$, and an upper bound on the maximum degree of the nodes in the network $\Delta$. Our second algorithm requires that each node know the topology of its two hop neighborhood and an upper bound on the size of the network $n$. Note that our algorithms and analysis only require that nodes know a good estimate of the values of the network parameters $n$ and $\Delta$, instead of their exact values. Such estimates are easy to obtain in many practical scenarios. For instance, consider the scenario where $n$ nodes with unit transmission radii are randomly placed in a square grid of area $n$. In this case, the maximum degree $\Delta = \Theta(\frac{\log n}{\log \log n})$ with high probability.

The remainder of this paper is organized as follows: in section 2, we introduce the basic models and definitions which will be used in the rest of the paper. We survey related work in section 3. In section 4, we present a simple centralized algorithm for constructing a WCDS in an ad hoc network. In section 5, we present our two distributed WCDS algorithms and their analysis. Section 6 and 7 deal with the applications of WCDS to broadcast and unicast routing and scheduling respectively. We present the results of our experimental performance evaluation in section 8. Section 9 contains conclusions and directions for future work.

## 2 Background

### 2.1 Network and Interference Model

We model the network connectivity using a unit disk graph (UDG) $G = (V, E)$: the nodes in $V$ are embedded in the plane. Each node has a maximum transmission range and an edge $(u, v) \in E$ iff $u$ and $v$ are within the maximum transmission range of each other. We assume that the maximum transmission range is the same for all nodes in the network (and hence w.l.o.g., equal to one unit).

Time is discrete and synchronous across the network; units of time are also referred to as time slots. Since the medium of transmission is wireless, whenever a node transmits a message, all its neighbors hear the message. If two or more neighbors of a node $w$ transmit at the same time, $w$ will be unable to receive any of those messages. In this case we also say that $w$ experiences collision. In any time slot, a node can either receive a message, experience collision, or transmit a message but cannot do more than one of these.

We work with the above interference model for ease of exposition and analysis. However, all the results presented in this paper (with the exception of Section 7) easily extend to the so called "protocol model" of interference also.

## 2.2 Definitions

We now describe the definitions and notations used in the rest of the paper. All the definitions below are with respect to the undirected graph $G = (V, E)$.

**Connected Dominating Set (CDS):** A set $W \subseteq V$ is a dominating set iff every node $u \in V$ is either in $W$ or is adjacent to some node in $W$. If the induced subgraph of the nodes in $W$ is connected, then $W$ is a connected dominating set (CDS). A Minimum Connected Dominating Set (MCDS) is a CDS with the minimum number of nodes.

**Maximal Independent Set (MIS):** A set $M \subseteq V$ is an independent set iff no two nodes in $M$ are adjacent to each other. $M$ is also a Maximal Independent Set (MIS) if there exists no set $M' \supseteq M$ such that $M'$ is an independent set. Note that, in an undirected graph, every MIS is a dominating set.

**Well Connected Dominating Set (WCDS):** A CDS $W$ is a WCDS if it satisfies the following properties:

**(P1) Low Size:** Let $OPT$ be an MCDS for $G$. Then, $|W| \leq k_1|OPT|$, where $k_1$ is a constant.

**(P2) Low Degree:** Let $G' = (W, E')$ be the graph induced by the nodes in $W$. For all $u \in W$, let $d'(u)$ denote the degree of $u$ in $G'$. Then, $\forall u \in W, d'(u) \leq k_2$, where $k_2$ is a constant.

**(P3) Low Stretch:** Let $D(p, q)$ denote the length of the shortest path between $p$ and $q$ in $G$. Let $D_W(p, q)$ denote the length of the shortest path between $p$ and $q$ such that all the intermediate nodes in the path belong to $W$. Let $s_W \doteq \max_{\{p,q\} \in V} \frac{D_W(p,q)}{D(p,q)}$. Then, $s_W \leq k_3$, where $k_3$ is a constant.

**Distance-k Neighborhood (Dk-neighborhood):** For any node $u$, the Dk-neighborhood of $u$ is the set of all other nodes which are within $k$ hops away from $u$.

**Distance-2 Vertex Coloring (D2-coloring):** D2-coloring is an assignment of colors to the vertices of the graph such that every vertex has a color and two vertices which are D2-neighbors of each other are not assigned the same color. Vertices which are assigned the same color belong to the same *color class*. This definition is motivated by the fact that nodes belonging to the same color class can transmit messages simultaneously without any collisions.

# 3 Related Work

CDS was first proposed as a candidate for virtual backbones in [10, 9, 23]. Guha and Khuller [13] studied MCDS and showed that computing an MCDS is NP-hard in arbitrary undirected graphs. They also presented centralized approximation algorithms which are guaranteed to produce a CDS of size $O(\log n)$ times that of an MCDS. Dubashi *et al.*[11] present a fast distributed CDS algorithm for undirected graphs with a running time of $O(\log^3 n)$. In addition, the solution produced by their algorithm has size $O(\log \Delta)$ times MCDS and stretch $O(\log n)$. However, this algorithm does not take into account the loss of messages due to collisions in wireless network. Although this algorithm is very attractive for general networks, it is not directly applicable to wireless ad hoc networks. The algorithms in [24, 27] construct a CDS of size $O(n)$ times MCDS and stretch $O(1)$ in arbitrary undirected graphs. The time complexity of the algorithm in [27] is $\Theta(\Delta^3)$. However, these algorithms also do not consider message losses due to collisions in their model.

It was shown in [8] that computing an MCDS is NP-hard even for UDGs. Cheng *et al.*[7] propose a centralized polynomial time approximation scheme (PTAS) for approximating MCDS in UDGs. Several distributed approximation algorithms exist for computing MCDS in UDGs [26, 18, 2, 3, 5]. These algorithms produce a solution whose size is within $O(1)$ times that of an MCDS. The time and message complexity of these algorithms are $O(n)$ and $O(n \log n)$ respectively. All these algorithms have a stretch of $O(n)$ [1]. Alzoubi *et al.*[4] proposed a distributed CDS algorithm for UDGs which has $O(n)$ time and message complexity and which results in a CDS of size $O(1)$ times MCDS. Recently, Alzoubi [1] showed that this CDS also has $O(1)$ stretch. We improve upon the time complexity of all the above algorithms by proposing the first sub-linear time distributed algorithms for ad hoc networks which constructs a WCDS of size $O(1)$ times MCDS and $O(1)$ stretch. In particular, we note that in comparison with [1], we achieve a drastic decrease in the time complexity (from $O(n)$ to $O(\log^2 n)$ at the expense of a slight increase in the message complexity (from $O(n)$ to $O(n \log n)$).

A basic primitive used in our first distributed algorithm is D2-coloring of the network. This primitive is well studied in the context of broadcast scheduling and channel assignment [22, 21, 19]. In [22], it was shown that even in the case of UDGs, it is NP-hard to minimize the number of colors used in the D2-coloring. However, for many restricted graph classes such as UDGs, several *centralized* approximation algorithms exist which use within $O(1)$ times the number of colors used by an optimal D2-coloring[22, 14, 20]. In this paper, we present the first distributed algorithm for D2-vertex coloring in UDGs with $O(\Delta \log^2 n)$ running time and which uses at most $O(1)$ times the number of colors used by an optimal D2-coloring.

Network-wide broadcasting is an important application for CDS in ad hoc networks. Several CDS based broadcasting algorithms exist where only the nodes in the CDS are involved in retransmitting the messages. Gandhi *et al.*[12] proposed a collision-free distributed broadcast scheme for UDGs, which broadcasts a set of offline messages. In this scheme, both the number of retransmissions and the broadcast latency is within O(1) times their optimal values. We improve upon this result in this paper by proposing a WCDS based online collision-free broadcast algorithm, which guarantees low latency, low number of retransmissions and high throughput, all within O(1) times their optimal values.

Luby[16, 17] proposed randomized distributed algorithms for vertex coloring and MIS construction in arbitrary undirected graphs. We adapt these algorithms for D2-coloring and MIS construction in UDGs. Luby's algorithms were originally meant for a system of parallel processors. Our adaptations which are meant for wireless ad hoc networks are complicated by the fact that messages can be lost due to collisions. Topkis [25] analyzed the time-complexity of flooding for all-to-all broadcasting in a wired network. We use some of the proof techniques from [25] in our analysis WCDS based broadcasting in Section 6.

Kumar *et al.*[15] propose algorithms for unicast packet scheduling under the D2-edge interference model. We utilize one of their results on end-to-end unicast scheduling to obtain our results in section 7. Our results in Section 7 are motivated by the work of Chen *et al.*[6], which proposes a distributed algorithm for constructing a forwarding backbone, which preserves the routing capacity of the underlying network.

# 4 A centralized WCDS algorithm

We now present a centralized algorithm for constructing a WCDS. Algorithm 1 takes as input a connected unit disk graph $G = (V, E)$. Lines 2 to 7 compute an MIS for this graph. This is done by iteratively choosing vertices which are currently not in MIS and which do not currently have a neighbor in MIS. Since $G$ is an undirected graph, any maximal independent set for $G$ is also a dominating set. Lines 8 to 11 connect the nodes in MIS. Specifically, every MIS node $u$ is connected to every other MIS node $v$ in its D3-neighborhood, using a shortest path between $u$ and $v$. Nodes in the shortest paths along with the nodes in MIS constitute the WCDS $W$. Alzoubi *et al.*[4] proposed a linear time distributed implementation of this algorithm. Alzoubi [1] proved that the CDS constructed using this algorithm satisfies properties **P1**, **P2** and **P3** introduced in Section 2 and hence is a WCDS.

**Algorithm 1** $CENTRALIZED-WCDS(G = (V, E))$

**Require:** $G$ is a connected unit disk graph

1: $U = V$
2: $MIS = \phi$
3: **while** $U \neq \phi$ **do**
4:     Pick any $u \in U$
5:     $MIS = MIS \cup \{u\}$
6:     $U = U \setminus (\{u\} \cup N(u))$
7: **end while**
8: $W = MIS$
9: **for all** $\{u, v\} \subseteq MIS$ **do**
10:     $P =$ set of nodes in the shortest path from $u$ to $v$
11:     $W = W \cup P$
12: **end for**
13: return $W$

# 5 Distributed Implementation

In this section, we describe two distributed algorithms for constructing WCDS. Both the algorithms are randomized and can be parametrized to yield the correct output upon termination with very high probability.

## 5.1 WCDS via D2-coloring

We now present our first distributed algorithm for WCDS. We assume that each node knows its neighbors, the maximum degree $\Delta$, and the size of the network $n$. There are three stages in the algorithm. The first stage involves D2-coloring of the nodes in the network using a list of $c$ colors.The second stage involves constructing a Maximal Independent Set (MIS) and the third stage involves connecting the MIS. The second and third stages can be easily implemented since transmissions can be scheduled using the D2-coloring computed in the first stage. We now present these stages in detail.

**Stage 1: D2-coloring** This stage is parametrized by three positive integers: $c$, $t$, and $r$ (to be specified later). Each node $u$ has a list of colors $L(u)$ which is initialized to $\{1, 2, \dots c\}$. Time is divided into frames of length $c$ time slots. The coloring algorithm proceeds in a synchronous round by round fashion. Typically, each round involves the following steps. Some of the yet-uncolored nodes choose a tentative colors for themselves. Some of these nodes will be successful, since none of their D2-neighbors would have chosen the same tentative color as themselves. In this case, the tentative color becomes the permanent color for these nodes. The unsuccessful nodes update their color list by removing the set of colors chosen by their successful D2-neighbors in this round and continue their attempts to color themselves in the future rounds. The coloring algorithm terminates after $t$ rounds. We now present the details of a specific round.

Each round consists of four phases: **TRIAL**, **TRIAL-REPORT**, **SUCCESS** and **SUCCESS-REPORT**. The details of these phases are given below.

**TRIAL:** Only the yet-uncolored nodes participate in this phase. This phase consists of a single frame. At the beginning of this phase, each yet-uncolored node $u$ *wakes up* or *goes to sleep* with probability 1/2 respectively. If $u$ is awake, it chooses a tentative color $color(u)$ uniformly at random from $L(u)$. Note that $L(u)$ is the list of colors available for node $u$ in the current round and this list may change in the future rounds. Node $u$ then transmits a TRIAL message $\{ID(u), color(u)\}$ at the time slot corresponding to $color(u)$ in this frame: for e.g., if $u$ is awake and if $color(u) = 5$, $u$ transmits the message $\{ID(u), 5\}$ at the fifth time slot of this frame. In general, the TRIAL message (and other types of messages below) may not reach all the neighbors of $u$ due to collisions.

**TRIAL-REPORT:** This phase consists of $r$ frames. At the beginning of this phase, *every* node $u$ in the network prepares a TRIAL-REPORT message. This message is the concatenation of all the TRIAL messages received by $u$ in this round. During *every* frame of this phase, $u$ chooses a time slot independently at random within the frame, and broadcasts the TRIAL-REPORT message during this time.

**SUCCESS:** This phase consists of a single frame. At the beginning of this phase, every node $u$ which is *awake*, determines if the tentative color it chose during the TRIAL phase is a safe color or not. Intuitively, $color(u)$ is safe iff no node in its D2-neighborhood chose the same color as $u$. In our algorithm, $u$ deems $color(u)$ to be safe if the following conditions hold:

1. $u$ received a TRIAL-REPORT message from each of its neighbors.
2. Each TRIAL-REPORT message received by $u$ contained the TRIAL message sent by $u$.

If the above conditions are met, $color(u)$ becomes the permanent color for $u$. In this case, $u$ creates a SUCCESS message $\{ID(u), color(u)\}$ and broadcasts it to all its neighbors. This transmission is done at the time slot corresponding to $color(u)$ within this frame. In future rounds, $u$ does not participate in the **TRIAL** and **SUCCESS** phases since it successfully colored itself in this round.

**SUCCESS-REPORT:** This phase is similar to the **TRIAL-REPORT** phase. The SUCCESS-REPORT message for *every* node $u$ in the network is a concatenation of SUCCESS messages which were received by $u$ in this round. This phase also consists of $r$ frames. During *every* frame of this phase, $u$ chooses a time slot independently at random within the frame and broadcasts its SUCCESS-REPORT message during this slot. Crucially, *at the end of this phase, any yet-uncolored node $v$ removes from its list $L(v)$, any color found in the* SUCCESS *or* SUCCESS-REPORT *messages received by $v$ in this round*. This ensures that, in the future rounds, $v$ does not choose the colors of its successful D2-neighbors.

This completes the description of a single round of the coloring stage. As mentioned earlier, the coloring stage consists of $t$ such rounds.

**Stage 2: Constructing the MIS** The previous stage ensures that all nodes in the network have a valid D2-coloring using colors $\{1, 2, \ldots c\}$. During this stage, a maximal independent set (MIS) is built iteratively in $c$ time slots. During slot $i$, all nodes belonging to color class $i$ attempt to join the MIS. A node joins the MIS if and only if none of its neighbors are currently part of the MIS. After joining the MIS, the node broadcasts a message to its neighbors indicating that it joined the MIS. Nodes transmitting during the same time slot belong to the same color class and hence do not share a common neighbor. Clearly, this stage requires exactly $c$ time steps.

**Stage 3: Connecting the MIS** This stage requires six phases. Each phase is one frame long and a single frame is of length $c$. As in stage two, nodes transmit only during the time slot corresponding to their D2-color. During the first phase, all MIS nodes transmit a PHASE-1 message. This message just consists of the node's ID. In the second phase, any node $u$ which received a PHASE-1 message, transmits a PHASE-2 message. This message is a concatenation of $ID(u)$ and all the PHASE-1 messages received by $u$. In the third phase, any node $u$ which received a PHASE-2 message, transmits a PHASE-3 message. This message is a concatenation of $ID(u)$ and all the PHASE-2 messages received by $u$.

By the end of the third phase, every MIS node $u$ knows every other MIS node $v$ in its D3-neighborhood. Node $u$ also knows all paths of length at most three between itself and $v$. Node $u$ constructs a PHASE-4 message as follows: for every other MIS node $v$ such that $v$ is in its D3-neighborhood and $ID(v) > ID(u)$, $u$ chooses a path of length at most three hops between itself and $v$. It adds this information to its PHASE-4 message. All MIS nodes transmit a PHASE-4 message during the fourth phase. Every node $u$ which recieved a PHASE-4 message transmits a PHASE-5 message. This message is a concatenation of all the PHASE-4 messages received by $u$. Finally every node $u$ which received a PHASE-5 message transmits a PHASE-6 message. This message is a concatenation of all the PHASE-5 messages received by $u$. By the end of this stage, any MIS node $u$ knows the path between itself and any other MIS node $v$ which in its D3-neighborhood. In addition, any node $w$ which is not part of the MIS, knows if it is part of the final WCDS or not.

This completes the description of our first distributed WCDS algorithm.

### 5.1.1 Analysis

Let the parameters in the algorithm have the following values: $c = k_1\Delta$, $t = k_2 \log n$, and $r = k_3 \log n$, where $k_1$, $k_2$ and $k_3$ are constants. We prove the following theorems pertaining the time and message complexity and the correctness of our algorithm.

**Theorem 5.1** *The running time of the algorithm is* $O(\Delta\log^2 n)$.

**Proof** The first stage consists of $t$ rounds, each of which consists of $2(r+1)$ frames. The second and third stages consist of 1 and 6 frames respectively. Hence, the total number of frames is $O(tr) = O(\log^2 n)$. All frames are of length $c = O(\Delta)$. Hence, the running time is $O(\Delta\log^2 n)$. ■

The following definition and claims are useful for the rest of the analysis.

**Definition:** Let $S$ be a set of disks on the plane. Let $C$ be a disk on the plane. For any disk $s$, we let $s$ denote both the disk and the set of points contained within the disk. We say that $S$ is a covering for $C$ iff the following hold:

1. $C \subseteq \bigcup_{s\in S} s$
2. $\forall s \in S, s \cap C \neq \phi$
3. $\forall\{s_1, s_2\} \subseteq S$, the center of $s_1$ lies outside the center of $s_2$ and vice versa.

We state the following simple claim from geometry.

**Claim 5.2** *Let $S$ be a set of disks of radius $r_1$ and $C$ be a disk of radius $r_2$. Let $S$ be a covering for $C$. Then, $|S| \leq k_0(\frac{r_1+r_2}{r_1})^2$, where $k_0$ is a constant. In addition, such a covering always exists.*

We use the above claim to prove the following lemma.

**Lemma 5.3** *Let $G = (V, E)$ be a UDG with maximum degree $\Delta$. Let $C$ be a disk of radius $r \geq \frac{1}{2}$. The number of nodes of $V$ which lie within $C$ is at most $4k_0r^2(\Delta+1)$.*

**Proof** Let $S$ be a set of disks of radius $\frac{1}{2}$ which cover $C$. The maximum number of nodes which lie within any disk in $S$ is at most $\Delta+1$ (since such nodes form a clique). The number of disks in $S$ by claim 5.2 is at most $4k_0r^2$. Hence, the lemma follows. ■

**Theorem 5.4** *All messages in the algorithm require at most $O(\Delta \log n)$ bits.*

**Proof** The TRIAL and SUCCESS messages transmitted by a node $u$ are of the form $\{ID(u), color(u)\}$. The PHASE-1 and PHASE-4 messages are of the form $\{ID(u)\}$. The PHASE-2 (PHASE-5) message is a concatenation of a node's ID and the PHASE-1 (PHASE-4) messages transmitted by its one-hop MIS neighbors. All one-hop MIS neighbors are within a disk of radius one and the subgraph induced by the MIS nodes is a UDG with maximum degree zero. Hence, by lemma 5.3, there can be at most $O(1)$ MIS nodes within a disk of radius 1. Hence PHASE-1, PHASE-2, PHASE-4, and PHASE-5 messages require at most $O(\log n)$ bits. All other messages are a concetanation of at most $\Delta$ messages of the previous types and hence require at most $O(\Delta \log n)$ bits. ■

**Theorem 5.5** *The total number of messages transmitted by the algorithm is at most $O(n \log^2 n)$.*

**Proof** There are $O(\log^2 n)$ frames in the algorithm. Each node transmits at most once per frame. Hence the theorem follows. ■

We note that the term w.h.p below implies "with high probability". Specifically, this probability is $1-\frac{1}{\delta^n}$, where $\delta$ is a constant which can be made arbitrarily high by choosing constants $k_1$, $k_2$ and $k_3$ appropriately. We now prove the following lemmas.

**Lemma 5.6** *Let $u$ be a node which is yet-uncolored at the beginning of a round $i$. Let $L_i(u)$ be the color-list of $u$ in the beginning of round $i$. Let $S$ be the set of D2-neighbors of $u$ which are yet-uncolored at the beginning of round $i$. Then, $|L_i(u)| \geq |S|+1$. In particular, the color-list of $u$ is never empty before $u$ is successfully colored.*

**Proof** Recall that $c = k_1\Delta$ is the initial size of the color-list for all nodes. Lemma 5.3 implies that the maximum number of D2-neighbors for any node is at most $16k_0(\Delta+1)$. Let the constant $k_1$ be chosen such that $k_1 > 32k_0$. Hence each node initially has a list of $c$ colors which is *strictly* greater than the number of its D2-neighbors. A node removes at most one color for each of its successful D2-neighbors in any round. Hence, the lemma follows. ■

The above lemma ensures that the TRIAL phase is well defined: i.e., no node has an empty color-list before it is successfully colored. The following lemma ensures that the TRIAL-REPORT and SUCCESS-REPORT phases are executed correctly, resulting in a valid D2-coloring of vertices.

**Lemma 5.7** *Let $u$ be a node which transmits TRIAL(SUCCESS)-REPORT messages during a fixed round $i$. Let $v$ be a fixed neighbor of $u$. Consider the event that $v$ does not receive even a single TRIAL(SUCCESS)-REPORT message collision-free from $u$ during round $i$. The probability of this event occuring is at most $\frac{1}{n^\delta}$.*

**Proof** Consider the event that the TRIAL-REPORT message transmitted by node $u$ in a particular frame is not received by node $v$. We now compute the probability of this event. Recall that $u$ (and any other node) chooses a time slot independently at random within a frame and transmits the TRIAL-REPORT message. Let $T_{x,j}$ denote the time slot chosen by a node $x$ in frame $j$. Let $S = \{v\} \bigcup N(v) \setminus \{u\}$. Note that $S$ is the set of nodes which could interfere with $u$'s transmission and cause collision at $v$. Specifically, node $v$ will not receive $u$'s message during frame $j$ only if there exists a node $x \in S$ such that $T_{x,j} = T_{u,j}$. We have,

$$\begin{aligned}
\Pr[\exists x \in S, T_{x,j} = T_{u,j}] &= 1 - \Pr[\forall x \in S, T_{x,j} \neq T_{u,j}] \\
&= 1 - \prod_{x \in S} \Pr[T_{x,j} \neq T_{u,j}] \\
&= 1 - \prod_{x \in S} \left(1 - \frac{1}{c}\right) \\
&= 1 - \left(1 - \frac{1}{c}\right)^{|S|} \\
&\leq 1 - \left(1 - \frac{1}{c}\right)^{\Delta}
\end{aligned}$$

Since $c = k_1 \Delta$, by letting $k_1 > 1$, we have

$$\begin{aligned}
\left(1 - \frac{1}{c}\right)^{\Delta} &\geq \frac{1}{4} \\
\text{Hence, } \Pr[\exists x \in S, T_{x,j} = T_{u,j}] &\leq \frac{3}{4}
\end{aligned}$$

In order for $v$ not to get even a single TRIAL-REPORT message of $u$, the above event ($\exists x \in S, T_{x,j} = T_{u,j}$) should occur for all frames $j \in \{1, \ldots, r\}$ in the **TRIAL-REPORT** phase. Thus,

$$\begin{aligned}
\Pr & \quad [v \text{ not receiving any TRIAL-REPORT from } u] \\
\leq & \quad \left(\frac{3}{4}\right)^r = \left(\frac{3}{4}\right)^{k_3 \log n} = \frac{1}{n^\delta}
\end{aligned}$$

Here, $\delta$ is a constant which can be made arbitrarily high by choosing an appropriate value of $k_3$. This completes the proof of the Lemma. ■

**Lemma 5.8** *Let $u, v$ be any fixed pair of D2-neighbors. When the algorithm terminates, $u$ and $v$ have the same color with at most a negligibly low probability of $\frac{1}{n^\delta}$, where $\delta$ is a constant which can be made arbitrarily high.*

**Proof** Assume that $u$ and $v$ have been successfully colored with the color $z$, during rounds $i$ and $j$ respectively. W.l.o.g., let $i \leq j$. There are two possible cases.

The first case occurs when $u$ and $v$ are neighbors of each other. In this case, since $u$ is colored successfully in round $i$, no neighbor of $u$ *and* no neighbor $v$ (except $u$), chose the tentative color $z$ during round $i$ (otherwise, the TRIAL-REPORT of $v$ would not have contained the TRIAL of $u$, and $u$ would not have deemed itself successful). This also implies that $v$ received the SUCCESS message of $u$ during round $i$ collision-free. Hence, $v$ would have removed color $z$ from its list during round $i$, leading to a contradiction.

The second case occurs when $u$ and $v$ are not neighbors of each other. In this case, there exists a node $x$ which is a neighbor of both $u$ and $v$. By the same arguments as above, $x$ receives the TRIAL and SUCCESS messages of $u$ during round $i$, collision-free. Hence, the TRIAL-REPORT and SUCCESS-REPORT messages of $x$ in round $i$, contains the color $z$. Node $v$ would deem $z$ as its permanent color during round $j = i$, only if it did not receive any TRIAL-REPORT messages transmitted by $x$ in round $i$. Node $v$ would choose color $z$ during a round $j > i$, only if it did not receive any SUCCESS-REPORT messages transmitted by $x$ in round $i$. By Lemma 5.7, both these events occur with the negligibly low probability of $\frac{1}{n^\delta}$, where $\delta$ is a constant which can be made arbitrarily high. This completes the proof of the lemma. ■

**Lemma 5.9** *Consider the final colors of all the nodes after the D2-coloring algorithm terminates. No two nodes which are D2-neighbors of each other have the same color, w.h.p.*

**Proof** There are at most $n^2$ pairs of neighbors. By Lemma 5.8 and using the union bound, the probability of any of these pairs having the same color is at most $n^2 \left(\frac{1}{n}\right)^\delta$. Hence, the probability of no pair of neighbors having the same D2-color is at least $1 - \frac{1}{n^{\delta-2}}$, where $\delta$ is a constant which can be made arbitrarily high. This completes the proof of the lemma. ■

**Lemma 5.10** *Let node $u$ be yet-uncolored at the beginning of a particular round $i$. Let $S(u,i)$ denote the event that $u$ was successfully colored during round $i$. Then,* $\Pr[\overline{S(u,i)}] \leq \frac{4}{5}$.

**Proof** Recall that $L_i(u)$ is the color-list of $u$ in the beginning of round $i$. Let $N$ be the set of D2-neighbors of $u$ which are yet-uncolored in the beginning of round $i$. Let $A(x)$ denote the event that node $x$ is *awake* during round $i$. Let $C(x,z)$ denote the event that node $x$ chose the tentative color $z$ during the TRIAL phase of round $i$. We note that for the event $\overline{S(u,i)}$ to occur, atleast one of the the following three events should occur:

1. $\overline{A(u)}$: $u$ was not awake in round $i$. This event occurs with probability $\frac{1}{2}$.

2. $\exists(x \in N, z \in L_i(u))$ such that $C(x,z)$ and $C(u,z)$: Some D2-neighbor of $u$ choose the same color as $u$. We denote this event as $F(u)$.

3. $u$ did not receive a TRIAL-REPORT message from all its neighbors. By Lemma 5.7, this event occurs with an arbitrarily low probability $\epsilon$.

We now compute $\Pr[F(u)]$. We first note that for any node $x$, the probability that it chooses any color from its list is at most 1/2, since it needs to be *awake* before choosing a color. In addition, $\Pr[C(x,z)] \leq \frac{1}{2|L_i(x)|}$, since $x$ chooses a color uniformly at random from its list. We have,

$$\begin{aligned}
\Pr[F(u)] &= \sum_{z \in L_i(u)} \Pr[C(u,z)](\sum_{x \in N} \Pr[C(x,z)]) \\
&= \frac{1}{2|L_i(u)|} \sum_{z \in L_i(u)} (\sum_{x \in N} \Pr[C(x,z)]) \\
&= \frac{1}{2|L_i(u)|} \sum_{x \in N} (\sum_{z \in L_i(u)} \Pr[C(x,z)]) \\
&\leq \frac{1}{2|L_i(u)|} \sum_{x \in N} \frac{1}{2} \\
&\leq \frac{|N|}{4|L_i(u)|}
\end{aligned}$$

By Lemma 5.6, $|L_i(u)| \geq |N| + 1$. Hence, the above probability is at most $\frac{1}{4}$. Hence, we have

$$\Pr[\overline{S(u,i)}] \leq \frac{1}{2} + \frac{1}{4} + \epsilon$$

where $\epsilon$ is an arbitrarily small constant. This completes the proof of the lemma. ■

**Lemma 5.11** *All nodes are successfully colored when the D2-coloring algorithm terminates w.h.p.*

**Proof** By Lemma 5.10, the probability of a particular node $u$ remaining yet-uncolored after the algorithm terminates is at most $\left(\frac{4}{5}\right)^t = \left(\frac{4}{5}\right)^{k_2 \log n} = \frac{1}{n^\delta}$. Hence, the probability of some node remaining yet-uncolored after the algorithm terminates is at most $\frac{n}{n^\delta} = \frac{1}{n^{\delta-1}}$. Here, $\delta$ is a constant which can made arbitrarily high by choosing the appropriate value of $k_2$. This concludes the proof of the lemma. ■

**Theorem 5.12** *The first stage computes a valid D2-coloring w.h.p.*

**Proof** Lemmas 5.9 and 5.11 together yield this theorem. ■

**Theorem 5.13** *The second and third stages compute a valid MIS and WCDS respectively w.h.p.*

**Proof** Observe that if the first stage produces a valid D2-coloring, then no packets are lost due to collision and the second and third stages are executed correctly. By Theorem 5.12, the first stage computes a valid D2-coloring w.h.p. Hence this theorem follows. ■

## 5.2 WCDS via. D2-topology

We now describe our second distributed implementation for WCDS. We assume that each node knows its D2-topology, i.e., the nodes in its D2-neighborhood and the edges between these nodes. This algorithm comprises of two stages. An MIS is constructed in the first stage and it is connected in the second stage. We present the details of these stages below.

**Stage 1: Constructing the MIS** This stage proceeds in a synchronous round by round fashion. The MIS is initially empty. Typically, some nodes are successful at the end of each round. A node is deemed successful if either the node joins the MIS or one of its neighbors joins the MIS. Successful nodes do not participate in the future rounds, while remaining nodes continue their attempts to be successful in the future rounds. The MIS construction terminates after $t$ rounds.

During this stage, each node $u$ maintains a status variable which is defined as follows: status($u$)=*in* iff $u$ has joined the MIS; status($u$)=*out* if any neighbor of $u$ has joined the MIS; status($u$)=*unsure* otherwise. All nodes are initially *unsure* and become *in* or *out* of MIS during the course of the algorithm. Let $V_i$ be the set of nodes whose status is *unsure* at the end of round $i-1$. For any node $u \in V_i$, let $N_i(u) = N(u) \cap V_i$. Let $MIS_i$ be the set of nodes which join MIS in round $i$.

There are four phases in each round of the first stage: **TRIAL**, **CANDIDATE-REPORT**, **JOIN**, and **PREPARE**. We now present the details of these phases for a particular round $i$.

**TRIAL:** In this phase, each *unsure* node decides if it is a candidate for $MIS_i$. Specifically, each *unsure* node $u$ chooses itself to be a candidate for joining $MIS_i$, with probability $\frac{1}{2(|N_i(u)|+1)}$. Node $u$ will not be a candidate in this round with the complement probability. This phase does not required any message transmissions.

**CANDIDATE-REPORT:** This phase ensures that each node knows if there is a neighbor who is a candidate. This step consists of $p$ time frames, each frame consisting of two slots. During *every* frame of this phase, each *candidate* node chooses one of the two slots independently at random and broadcasts a CANDIDATE message. Any node which receives a CANDIDATE message or expreriences collision during this phase, knows that there is a neighboring candidate; otherwise it assumes that there is no neighboring candidate.

**JOIN:** This phase requires a single time slot. In this phase, some *unsure* nodes become either *in* or *out*. How should a candidate decide if it should join $MIS_i$ (become *in*)? A candidate joins $MIS_i$ if none of its neighbors are candidates for $MIS_i$, i.e., if it did not receive a CANDIDATE message during the previous phase. All nodes who joined $MIS_i$ transmit a JOIN message. *unsure* nodes which receive a JOIN message or experience collision, change their status to *out*. Other *unsure* nodes do not change their status.

**PREPARE:** Each *unsure* node $u$ computes $N_{i+1}(u)$ at the end of this phase. This phase consists of $p$ time frames. Each frame is further subdivided into $\alpha$ sub-frames of length $c$. During *every* frame of this phase, each node in $MIS_i$, chooses independantly at random, one of the $\alpha$ sub-frames. During this sub-frame, it broadcasts a PREPARE message using the algorithm in [12] to its D2-neighbors. The length of the sub-frame, $c$ is the number of time steps required by [12] to transmit a message from a node to its D2-neighbors. The PREPARE message broadcast by a node simply consists of its ID. By the end of this phase, every *unsure* node knows all the nodes in its D2-neighborhood which joined $MIS_i$. Hence, it can easily compute $N_{i+1}(u)$.

**Stage 2: Connecting the MIS**.
In this phase the MIS computed in the previous stage is connected using intermediate nodes. Specifically, every MIS node connects itself to every other MIS node which is at most three hops away. This stage consists of two phases: **HELLO** and **CONNECT**. We now present the details of these phases. For ease of exposition, we assume that each node knows its D3-topology, although the algorithm can be modified easily to such that the nodes know only their D2-topology.

**HELLO:** This phase is similar to the **PREPARE** phase in the first stage. The objective of this phase is for each MIS node to announce itself to other MIS nodes in its D3-neighborhood. This phase consists of $p$ frames, where each frame is subdivided into $\alpha'$ sub-frames of length $c'$. During *every* frame of this phase, each node $u \in MIS$ node selects independently at random, one of the $\alpha'$ sub-frames. During this sub-frame, $u$ broadcasts a HELLO message using the algorithm in [12] to its D3-neighborhood. By the end of this phase, each MIS node knows any other MIS node in its D3-neighborhood.

**CONNECT:** This phase is similar to the **HELLO** phase. The only difference arises in the contents of the CONNECT message. Each node $u \in MIS$ prepares its CONNECT message as follows. For every node $v \in MIS$ such that $v$ is in its D3-neighborhood and $ID(v) > ID(u)$, the CONNECT message of $u$ contains the tuple $\{ID(v), u \rightsquigarrow v\}$. $u \rightsquigarrow v$ is the shortest path between $u$ and $v$. As mentioned earlier, CONNECT messages are broadcast in the same way as the HELLO messages. Intermediate nodes, which are not part of the MIS, might join the WCDS, since they might be a part of the shortest path between two MIS nodes.

This completes the description of our distributed WCDS algorithm.

### 5.2.1 Analysis

Let $p = k_1 \log n$ and $t = k_2 \log n$. Let $\alpha$ and $\alpha'$ be the maximum number of MIS nodes in the D2 and D3 neighborhoods of any node respectively. Let $c$ and $c'$ be the number of time slots required by the algorithm in [12] to broadcast a message from any node to its D2 and D3 neighborhoods. We note that $\alpha$, $\alpha'$, $c$ and $c'$ are all fixed constants. We now state the following claims. We omit the proofs of most of the claims below due to lack of space.

**Theorem 5.14** *The running time of the algorithm is* $O(\log^2 n)$.

**Proof** The first stage consists of $t$ rounds, each of which consists of $(2p + 1 + p\alpha c)$ slots. The second and third stages together consist of $2p$ frames, each of which consists of $(\alpha' c')$ slots. Since $\alpha$, $\alpha'$, $c$, and $c'$ are constants, the running time is $O(tp) = O(\log^2 n)$. ■

**Theorem 5.15** *All messages transmitted in the algorithm require at most* $O(\log n)$ *bits.*

**Proof** The CANDIDATE, JOIN and PREPARE messages just consist of a node's ID. The HELLO and CONNECT messages transmitted from an MIS node $u$ to another MIS node $v$ just consists of ID's of $u$, $v$, and at most 2 intermediate nodes in the path between $u$ and $v$. Hence all these messages require at most $O(\log n)$ bits. ■

The following lemmas pertain to the correctness of the various phases in the algorithm.

**Lemma 5.16** *Consider a candidate node u during a particular round i of the algorithm. Consider a neighbor v of u. Node v either receives a CANDIDATE message or experiences collision w.h.p. during the* ***CANDIDATE-REPORT*** *phase of this round.*

**Proof** Recall that the **CANDIDATE-REPORT** phase consists of $p$ frames each consisting of two time slots. Let $A(u, v, j)$ denote the event that both $u$ and $v$ chose the same time slot to transmit their CANDIDATE messages during frame $j$. Node $v$ will not receive $u$'s CANDIDATE message and not experience collision during frame $j$ only if $A(u, v, j)$ occurs. If $v$ is not a *candidate* during round $i$, then $\Pr[A(u, v, j)] = 0$. Else, $\Pr[A(u, v, j)] = \frac{1}{2}$. Node $v$ will not receive $u$'s CANDIDATE message or experience collision during each of the $p$ frames in the phase, if the event $\bigwedge_{j=1}^{p} A(u, v, j)$ occurs.

$$\begin{aligned}\Pr[\bigwedge_{j=1}^{p} A(u,v,j)] &\leq \frac{1}{2^p} \\ &= \frac{1}{2^{k_1 \log n}} \\ &= \frac{1}{n^\delta}\end{aligned}$$

Here, $\delta$ can be made arbitrarily high by choosing an appropriate value of $k_1$. ■

**Lemma 5.17** *Consider a round i and a node $u \in MIS_i$. Consider a D2-neighbor v of u. v receives (atleast one of) the PREPARE message transmitted by u collision-free w.h.p. during the* ***PREPARE*** *phase of round i.*

**Lemma 5.18** *Consider a node $u \in MIS_i$ and a D3-neighbor v of u. Node v receives (atleast one of) the HELLO (CONNECT) messages transmitted by u collision-free w.h.p. during the* ***HELLO (CONNECT)*** *phase of the second stage.*

**Lemma 5.19** *Consider the following bad events:*

1. *During the* ***CANDIDATE-REPORT*** *phase of some round of the first stage, a neighbor of some candidate node did not receive any of the CANDIDATE messages transmitted by the candidate node and did not experience collision.*
2. *During some round of the algorithm, a D2-neighbor of some MIS node did not receive any of the PREPARE messages transmitted by the MIS node.*
3. *During some* ***HELLO (CONNECT)*** *phase of the second stage, a D3-neighbor of some MIS node did not receive any of the HELLO (CONNECT) messages transmitted by the MIS node w.h.p.*

*W.h.p., none of the above bad events happen during the course of the algorithm.*

**Proof** The proof for all three bad events stated above are similar and uses the union bound. By Lemma 5.16, during a fixed round $i$, for a fixed *candidate* node $u$ and for a fixed neighbor $v$ of $u$, the probability of $v$ not receiving any of $u$'s CANDIDATE messages and not experiencing collision is during round $i$ is at most $\frac{1}{\delta}$. There are $O(n^2)$ pairs of neighbors and $O(\log n)$ rounds. The probability of the bad event occuring during at least once during any of these rounds for any pair of neighbors is at most $O(\frac{n^2 \log n}{n^\delta}) = O(\frac{1}{n^\beta})$, where $\delta$ (and hence $\beta$) can be made arbitrarily high. Similarly, by lemmas 5.17 and 5.18, and by arguments which are essentially the same as the above, the second and third bad events occur with probability at most $O(\frac{1}{n^\beta})$, where $\beta$ can be made arbitrarily high. Hence, none of the three bad events occur w.h.p. ■

**Lemma 5.20** *The MIS computed at the end of the first stage is an independent set w.h.p.*

**Proof** The MIS computed at the end of the first stage will not be an independent set, only if one of the first two bad events in Lemma 5.19 occur. These events occur with negligible probability. Hence the lemma follows. ■

**Lemma 5.21** *If MIS is a maximal independent set, then the second stage computes a valid WCDS w.h.p.*

**Proof** Assume that the first stage computes an MIS which is a valid maximal independent set. Then the WCDS computed in the second stage will not be valid, only if the third bad event in Lemma 5.19 occurs. Lemma 5.19 states that this event does not occur during the second stage, w.h.p. Hence the lemma follows. ■

Let $H$ be a disk of radius $\frac{1}{2}$. Let $V(H)$ denote the set of nodes which lie within $H$. Consider a fixed round $i$ during the first stage of the algorithm. Let $C(u, i)$ denote the event that node $u$ was a *candidate* for $MIS_i$. Recall that $V_i$ is the set of *unsure* nodes before round $i$. Let $X_i(H)$ be the random variable which denotes the number of *candidate* nodes within disk $H$. Let $V_i(H) \doteq V(H) \bigcap V_i$. The following lemmas hold.

**Lemma 5.22** $\Pr[X_i(H) > 0] \leq \frac{1}{2}$

**Proof** Since $H$ is a disk of radius $\frac{1}{2}$, any two nodes within $H$ are neighbors of each other. Hence,

$$\begin{aligned}
\mathbf{E}[X_i(H)] &= \Sigma_{u\in V_i(H)} \Pr[C(u,i)] \\
&= \Sigma_{u\in V_i(H)} \frac{1}{2(|N_i(u)|+1)} \\
&\leq \Sigma_{u\in V_i(H)} \frac{1}{2(|V_i(H)|)} \\
&= \frac{1}{2}
\end{aligned}$$

■

**Lemma 5.23** *Let $v \in V_i(H)$ be a candidate for $MIS_i$. Then,* $\Pr[X_i(H) \geq 2|C(v,i)] \leq \frac{1}{2}$. *Hence,* $\Pr[X_i(H) = 1|C(v,i)] \geq \frac{1}{2}$. *Since this holds for any $v \in V_i(H)$,* $\Pr[X_i(H) = 1|X_i(H) \geq 1] \geq \frac{1}{2}$.

For the rest of the analysis, let $S$ be a set of disks of radius $\frac{1}{2}$ which cover the disk of unit radius centered at $u$. Let $H \in S$ contain node $u$.

**Lemma 5.24** $\Pr[u \in MIS_i|C(u,i)] \geq \beta$, *where $\beta > 0$ is a constant.*

Consider the graph $F$ whose vertices are the disks in $S$, and two vertices $H_1, H_2$ are adjacent in $F$ iff there exists nodes $u \in V(H_1)$ and $v \in V(H_2)$ such that $(u,v) \in E$. Claim 5.2 implies that the maximum degree of any node in $F$ is at most a constant $\psi$. We now construct a directed forest $\mathcal{F} = (S, I)$ as follows. For every $H_1 \in S$, let $p(H_1) = \min_{u\in V_i(H_1)} \Pr[C(u,i)]$. If $p(H_1) < \frac{1}{4|V_i(H_1)|\psi^2}$, then $|N_i(u)| \geq 2|V_i(H_1)|\psi^2$. Hence, there exists $H_2$ adjacent to $H_1$ in $F$ such that $|V(H_2) \bigcap N_i(u)| \geq 2|V_i(H_1)|\psi$. We add the edge $(H_2, H_1)$ in $\mathcal{F}$. Note that we add at most one in-edge for every node $H \in S$ and if the directed edge $(H_2, H_1)$ exists in $\mathcal{F}$, then $|V_i(H_2)| > |V_i(H_1)|$. Hence $\mathcal{F}$ is an out-directed forest. We now state the following claims.

**Claim 5.25** *Let $H$ be a vertex in $\mathcal{F}$ and let $Children(H)$ be the children of $H$ in $\mathcal{F}$. Then,* $|V_i(H)| \geq 2(\Sigma_{H'\in Children(H)}|V_i(H')|)$.

**Claim 5.26** *Let $\mathcal{T}$ be a tree in the forest $\mathcal{F}$. Let $H$ be any node in $\mathcal{T}$. Let $Desc(H)$ denote the descendents of $H$ in $\mathcal{T}$.* $|V_i(H)| \geq \Sigma_{H'\in Desc(H)}|V_i(H')|$. *In particular, this claim holds for the root of the tree $\mathcal{T}$.*

**Claim 5.27** *Let $\mathcal{T}$ be any tree in the forest $\mathcal{F}$ and let $H$ denote the root of $\mathcal{T}$.* $\Pr[|X_i(H)| > 0] \geq \gamma$, *where $\gamma > 0$ is a constant.*

Let $Z_i \subseteq V_i$ denote the set of nodes which are *candidates* for $MIS_i$ or which have a neighbor who is a *candidate* for $MIS_i$.

**Lemma 5.28** $\mathbf{E}[|Z_i|] \geq \kappa|V_i|$, *where $\kappa > 0$ is a constant.*

Let $\epsilon$ be the minimum probability of a fixed *candidate* in a round $i$ joining $MIS_i$. Lemma 5.24 ensures that $\epsilon > 0$ is atleast a constant. Let $succ_i \in V_i$ be the successful nodes during round $i$, i.e., these set of nodes either joined $MIS_i$ or have a neighbor which joined $MIS_i$. The following lemma holds.

**Lemma 5.29** $\mathbf{E}[|succ_i|] \geq \epsilon|V_i|$, *where $\epsilon > 0$ is a constant. Hence, for all $i > 0$,* $\mathbf{E}[|V_i|] \leq (1-\epsilon)\mathbf{E}[|V_{i-1}|]$.

**Theorem 5.30** *The expected number of messages transmitted during the algorithm is $O(n \log n)$.*

**Proof** During the second stage, each MIS node broadcasts atmost $2p$ messages to its D3-neighborhood. Each of these broadcasts involve $O(1)$ transmissions [12]. Hence, the total number of transmissions during this stage is at most $O(p|MIS|) = O(n \log n)$. We now compute the expected number of messages broadcast during the first stage. During the **PREPARE** phase in round $i$, each node in $MIS_i$ broadcasts $p$ messages to its D2 neighborhood. Each of these broadcasts involve $O(1)$ transmissions. Since each MIS node broadcasts in the **PREPARE** phase of a single round, the total number of messages transmitted during this phase is $O(p|MIS|) = O(n \log n)$. The total number of messages transmitted during the **JOIN** phase is $|MIS| = O(n)$. Finally, during the **CANDIDATE-REPORT** phase of a round $i$, all nodes in $X_i \subseteq V_i$ transmit $p$ candidate messages. Since $|V_i|$ decreases geometrically in expectation with each round, the total expected number of messages transmitted during this phase is at most $p\Sigma_i\mathbf{E}[|V_i|] = O(p|V_1|) = O(n \log n)$. This completes the proof of the theorem. ■

**Lemma 5.31** *All nodes are successful at the end of the first stage w.h.p.*

**Proof** The expected number of *unsure* nodes at the end of $p$ rounds is $\mathbf{E}[|V_{p+1}|]$. $\Pr[|V_{p+1}| > 0] \leq \mathbf{E}[|V_{p+1}|]$. By Lemma 5.29,

$$\begin{aligned}
\Pr[|V_{p+1}| > 0] &\leq \mathbf{E}[|V_{p+1}|] \\
&\leq |V_1|(1-\epsilon)^p \\
&\leq n(1-\epsilon)^{k_1 \log n} \\
&\leq \frac{n}{n^{\delta+1}} \\
&\leq \frac{1}{n^\delta}
\end{aligned}$$

where $\delta > 0$ is a constant which can be made arbitrarily large by choosing an appropriate value of $k_1$. Hence the lemma follows. ■

**Theorem 5.32** *The MIS computed by the first stage is valid w.h.p.*

**Proof** By Lemma 5.19, MIS is an independent set w.h.p. Lemma 5.31 all nodes are successful at the end of the first stage w.h.p and hence the MIS computed is maximal w.h.p. Hence the theorem follows. ■

**Theorem 5.33** *The WCDS computed at the end of the second stage is valid with high probability.*

**Proof** Theorem 5.32 implies that the MIS is valid w.h.p. Lemma 5.21 implies that if the MIS is valid, then the WCDS is valid w.h.p. Hence the theorem follows. ■

# 6 Network-wide Broadcasting

We now present a simple, online, collision-free, distributed algorithm for broadcasting messages across the network using a WCDS $W$. For ease of analysis, we assume that messages are generated only by nodes in $W$. Our algorithm requires that nodes in $W$ have a valid D2-coloring. Specifically, let $G'$ be the induced subgraph of nodes in $W$. The maximum degree of any node in $G$' is a constant (follows from **(P2)**, section 2.2). Hence, it is possible to D2-color all nodes in $W$ using only $k$ colors, where $k$ is a constant. This pre-processing can be done using the D2-coloring stage of the algorithm in section 5.1. We let $0, 1, ..., k-1$ denote the set of colors and $color(u)$ denote the color of node $u$.

Our broadcast scheme requires that each message has a unique sequence number associated with it. This assumption is needed in order to enforce a total ordering of broadcast messages generated in the network. This can be easily ensured in practice by labelling each message $\mu$ with a triplet $\langle T(\mu), S(\mu), LSN(\mu)\rangle$. Here, $T(\mu)$ is the time at which message $\mu$ was generated. $S(\mu)$ is the source of message $\mu$. $LSN(\mu)$ is a locally generated sequence number for message $\mu$ at its source $S(\mu)$. It is easy to verify that this labeling is sufficient to enforce a total ordering of the messages.

Let time be divided into frames of length $k$. Let $Q(\mu)$ denote the frame during which message $\mu$ was generated, i.e., $Q(\mu)k \leq T(\mu) < (Q(\mu)+1)k$. Let $0, 1, \ldots$ be the (totally ordered) set of broadcast messages generated in the network. Since the network is multi-hop, intermediate nodes need to assist in the broadcast operation by retransmitting the messages. In our scheme, only nodes in $W$ retransmit messages. Every node $u \in W$ retransmits every message exactly once. Let $R(u,q)$ denote the set of messages which have been received by node $u$ by the end of frame $q$. Node $S(\mu)$ is deemed to have received message $\mu$ during frame $Q(\mu)$. Let $X(u,q)$ denote the set of messages which have been retransmitted by node $u$ by the end of frame $q$. The following simple rule specifies the behaviour of any node $u \in W$ during a particular frame $q$.

- Let $P = R(u, q-1) \setminus X(u, q-1)$. If $P$ is empty, then $u$ does not retransmit any message in frame $q$. Otherwise, let $\lambda = \min(P)$ be the least numbered message in $P$. Node $u$ transmits message $\lambda$ in the time slot corresponding to $color(u)$ in frame $q$.

This simple scheme guarantees that all nodes in the network receive all messages collision-free. In addition, this scheme optimizes the latency, the number of retransmissions, and the throughput of the broadcast to within a constant factor of their respective optimal values.

## 6.1 Analysis

Imagine an "adversary" who generates broadcast messages in the network. The long term rate at which the adversary can generate messages is at most a constant, although he can generate messages in bursts. Specifically, let $G'$ denote the induced subgraph of $W$, and let $R'$ denote the diameter of $G'$. Recall that $k$ is the number of colors required for D2-coloring the WCDS. During any continuous window of length $2kR'$ time slots, our adversary is allowed to generate at most $R'$ messages. Thus the long term term message generation rate of our adversary is at most $\frac{1}{2k}$, which is a constant. Note that no broadcast protocol can support more than a unit long term rate, at most one message can be received by a receiver at every time slot. We now state the following theorem under this adversarial behaviour. We omit the proof for the theorem, due to lack of space.

**Theorem 6.1** *The latency experienced by any message $\mu$ is $O(R)$. This is at most $O(1)$ the latency experienced by the message in an optimal broadcast algorithm since any algorithm incurs a latency of $R$. All messages are received collision-free by all nodes in the network. In addition, the number of retransmissions for any message is at most $O(1)$ times the optimal number of retransmissions required to broadcast the message.*

# 7 Unicast Routing

In this section, we show that WCDS is an efficient backbone for unicast routing in ad hoc networks. We derive our results in this section under the Distance-2 *edge* interference model (D2-model) [19, 20, 15].

## 7.1 Distance-2 *Edge* Interference Model (D2-model)

The D2-model is motivated by MAC protocols such at 802.11, where a single transmission along an edge involves the transmission of the data from one end point and the transmission of an acknowledgement from the other end point. Thus both the end points behave as senders and receivers (of data and ack) during a single transmission along an edge. This is in contrast with the scenarios encountered so far where in each transmission, only one end point of the edge behaves as a sender and the other behaves as a receiver.

Specifically, let $(u, v)$ and $(p, q)$ be edges in the network. We say that $(p, q)$ interferes with $(u, v)$ iff $p$ or $q$ is a neighbor of $u$ or $v$. A transmission along the edge $(u, v)$ is considered collision-free iff there is no other transmission along any edge $(p, q)$ which interferes with $(u, v)$.

## 7.2 Efficient Backbones for Unicast Routing

Two critical components for any routing algorithm are its path selection and scheduling strategies. The path selection strategy determines the path along which each packet traverses in the network. The scheduling strategy determines the time at which a packet is transmitted along each edge on its path. The scheduling component also ensures that the packets are transmitted collsion-free (one may think of these components as the network and the MAC layer protocols respectively). Together, they uniquely determine the latency experienced by every packet in the network.

Our goal here is not to present algorithms for path selection or scheduling. Instead, we show that any routing algorithm could be modified to operate over a WCDS. Crucially, the modified routing algorithm will use only the nodes in WCDS as intermediate nodes in the paths, *without incurring significant loss in the quality of the paths and schedules* when compared with the original algorithm. We formalize this intuition below.

Let $\mathcal{P} = \{p_1, \dots p_n\}$ be a set of paths such that the maximum length of any path is $d$. We will refer to the elements of $\mathcal{P}$ as both paths and packets interchangeably. For any disk $z$, let $n(z)$ denote the number of edges in all the paths in $\mathcal{P}$ with an end point inside $z$. Let $Z$ be the set of all disks on the plane with radius 1/2. Let $c = \max_{z|z \in Z} n(z)$: i.e., $c$ is the maximum number of edges in $\mathcal{P}$ which have an end point inside any fixed disk of radius 1/2. We call $d$ and $c$, the *dilation* and *contestion* of $\mathcal{P}$ respectively. A schedule $S$ for $\mathcal{P}$ specifies the time at which every packet is transmitted collision-free along each edge in its path. The length of the schedule $|S|$ is the maximum latency of any packet in this schedule, i.e., the maximum time at which any packet traverses any edge. Observe that, under the D2-model, both $c$ and $d$ (and hence $\frac{c+d}{2}$) are lower bounds on the length of any schedule for $\mathcal{P}$. We now state the following surprising claim from [15].

**Claim 7.1** *Let $OPT$ be an optimal collision-free schedule for $\mathcal{P}$ under the D2-model. Let $|OPT|$ denote the length of $OPT$ (which is the maximum latency experienced by a packet in $OPT$). Then, $|OPT| = \Theta(c + d)$.*

This result implies that there always exists a schedule in which all packets reach their destinations collision-free such that the maximum latency experienced by any packet is at most $\Theta(c + d)$ (and not $\Theta(cd)$).

We now construct a new set of paths $\mathcal{P}'$ from $\mathcal{P}$ as follows. Consider any path $p = (u = u_0 \to u_1 \to u_2 \dots u_l = v)$ in $\mathcal{P}$. W.L.O.G., let $u$ and $v$ not belong to $W$. For any node $x$, let $dom(x)$ denote the dominator of $x$ in $W$. For each $p \in P$, we create $p' \in \mathcal{P}'$ such that $p' = (u = u_0 \to dom(u_0) \rightsquigarrow dom(u_1) \rightsquigarrow dom(u_2) \dots dom(u_l) \to u_l = v)$. Here $dom(a) \rightsquigarrow dom(b)$ is the path of length at most 3 from $dom(a)$ to $dom(b)$ in $W$. Let $d$ and $d'$ be the dilation of $\mathcal{P}$ and $\mathcal{P}'$ respectively. Let $c$ and $c'$ be the congestion of $\mathcal{P}$ and $\mathcal{P}'$ respectively. We now state the following theorem without proof (which we omit due to lack of space).

**Theorem 7.2** $c' + d' = \Theta(c + d)$.

This theorem states that the new set of paths $\mathcal{P}'$ through the unicast routing backbone, does not experience significantly more congestion or dilation than the original set of paths $\mathcal{P}$. Thus, by claim 7.1, there exists a schedule $S'$ in which every packet reaches from its source to its destination collision-free with a maximum latency of $\Theta(c' + d') = \Theta(c + d)$.

Intuitively, this result implies that any a schedule for a set of packets in the original network can be converted into a new schedule for this set of packets on top of the WCDS. The new schedule will use only the nodes in the WCDS for routing. However, the latency experienced of the new schedule is at most $O(1)$ times the latency of the original schedule. In this sense, the WCDS preserves the capacity of the underlying network.

# 8 Simulations

This section deals with the experimental evaluation of the WCDS constructed by our algorithms through simulations. The focus of our experiments is to study the structural properties (size, degree, and stretch) of the WCDS. In all the experiments, the network nodes were assigned unit transmission ranges and were placed uniformly at random within a square. The parameters which were varied in these experiments are the dimensions of the square $l$ and the density of the nodes. The former essentially dictates the diameter of the underlying graph, whereas the latter determines the degree. All the experiments were performed on strongly connected graphs. All data points were averaged over 10 simulation runs.

## 8.1 Observations

Figure 1 plots the size of the WCDS as a function of node density for various values of the square dimension $l$. For a given value of $l$, the fractiona number of nodes in the WCDS decreases as a function of the density. Observe that the size of the WCDS produced by our algorithms is proportional the size of the Maximal Independent Set (MIS) which is produced in the intermediate stage. Beyond a certain threshold density (which guarantees network connectivity w.h.p), the size of the MIS depends only upon the $l$ and not the node density. Hence, the fractional number of nodes in the WCDS is inversely proportional to the density, which explains the trend.

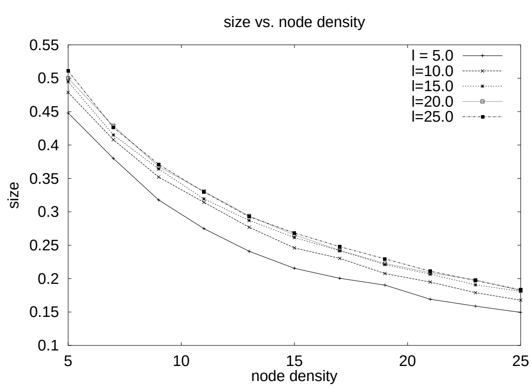


**Figure 1**: Size vs. Density: size represents the fractional number of nodes in the network that are part of the WCDS

Figure 2 plots the degree as a function of node density for various values of $l$. The degree plotted is the maximum degree of any node in the subgraph induced by the nodes in the WCDS. Notice how it increases as a concave function of the density (our analysis guarantees that the degree is upper-bounded by a constant).

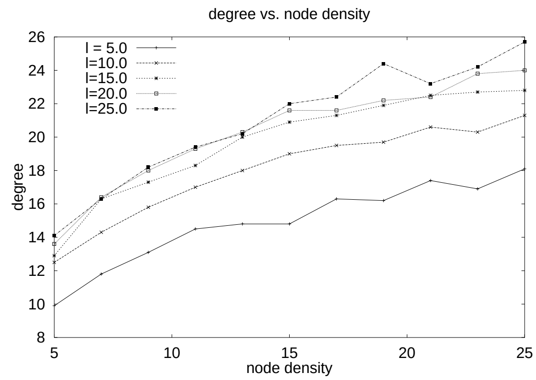


**Figure 2**: Degree vs. Node density

Figure 3 plots the estimated average stretch as a function of node density for various values of $l$. Recall that stretch was defined to be the maximum ratio between the pathlength of a pair of nodes through the WCDS to the pathlength in the original network. Although the exact value of stretch can be obtained in polynomial time, its exact computation takes time that is cubic in the size of the network and hence computationally infeasible. Hence, we estimate the stretch of the WCDS by first choosing 25 random source-destination pairs, and then averaging their stretch ratios. Clearly, the stretch is observed to be relatively invariant as a function of the node density.

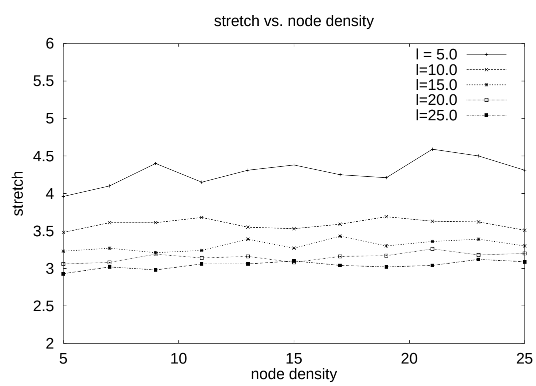


**Figure 3**: Stretch vs. Node density

# 9 Conclusions and Future Work

We presented fast, sub-linear time, randomized, distributed algorithms for connected dominating sets for ad hoc wireless networks. We also introduced a class of CDSs called Well Connected Dominating Sets (WCDS) which has many useful structural properties. These properties make WCDS the ideal backbone for broadcast and unicast routing in ad hoc wireless networks.

Several interesting research directions exist. The choices made during the design of the CDS has a great impact on its structural properties such as size, degree, stretch and so on. These properties could be crucial for applications which make use of the CDS. Currently, no theoretical or experimental analysis exists which explains how the design choices affect the structural properties of the CDS. We believe that such a study would be of immense value in practice. Another area of research is exploring the use of WCDS for tracking in sensor networks.

# References


[1] K. M. Alzoubi. Connected dominating set and its induced position-less sparse spanner for mobile ad hoc networks. In *Proceedings of the Eighth IEEE Symposium on Computers and Communications*, June 2003.

[2] K. M. Alzoubi, P.-J. Wan, and O. Frieder. Distributed heuristics for connected dominating sets in wireless ad hoc networks. *IEEE ComSoc/KICS Journal on Communication Networks*, 4:22–29, 2002.

[3] K. M. Alzoubi, P.-J. Wan, and O. Frieder. Message efficient construction of nontrivial connected dominating sets in wireless ad hoc networks. *To appear in Special Issue of ACM Journal of Monet*, 2002.

[4] K. M. Alzoubi, P.-J. Wan, and O. Frieder. Message-optimal connected-dominating-set construction for routing in mobile ad hoc networks. In *Proceedings of the Third ACM International Symposium on Mobile Ad Hoc Networking and Computing*, June 2002.

[5] K. M. Alzoubi, P.-J. Wan, and O. Frieder. New distributed algorithm for connected dominating set in wireless ad hoc networks. *IEEE HICSS35*, 2002.

[6] Benjie Chen, Kyle Jamieson, Hari Balakrishnan, and Robert Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 85–96. ACM Press, 2001.

[7] X. Cheng, X. Huang, D. Li, and D.-Z. Du. Polynomial-time approximation scheme for minimum connected dominating set in ad hoc wireless networks. Technical report.

[8] B. Clark, C. Colbourn, and D. Johnson. Unit disk graphs. *Discrete Mathematics*, 86:165–177, 1990.

[9] B. Das, R. Sivakumar, , and V. Bharghavan. Routing in ad-hoc networks using a virtual backbone. In *6th International Conference on Computer Communications and Networks (IC3N '97)*, pages 1–20, September 1997.

[10] Bevan Das and Vaduvur Bharghavan. Routing in ad-hoc networks using minimum connected dominating sets. In *ICC (1)*, pages 376–380, 1997.

[11] Devdatt Dubhashi, Alessandro Mei, Alessandro Panconesi, Jaikumar Radhakrishnan, and Arvind Srinivasan. Fast distributed algorithms for (weakly) connected dominating sets and linear-size skeletons. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 717–724. Society for Industrial and Applied Mathematics, 2003.

[12] Rajiv Gandhi, Srinivasan Parthasarathy, and Arunesh Mishra. Minimizing broadcast latency and redundancy in ad hoc networks. In *Proceedings of the fourth ACM international symposium on Mobile ad hoc networking and computing*, pages 222–232. ACM Press, 2003.

[13] Sudipto Guha and Samir Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, pages 374–387, 1998.

[14] Sven O. Krumke, Madhav V. Marathe, and S. S. Ravi. Models and approximation algorithms for channel assignment in radio networks. *Wireless Networks*, 7(6):575–584, 2001.

[15] V. S. Anil Kumar, Madhav V. Marathe, Srinivasan Parthasarathy, and Aravind Srinivasan. End-to-End Packet Scheduling in Ad Hoc Networks. To appear in ACM SODA 2004.

[16] Michael Luby. A simple parallel algorithm for the maximal independent set problem. *SIAM Journal on Computing*, 15(4):1036–1053, November 1986.

[17] Michael Luby. Removing randomness in parallel computation without a processor penalty. *Journal of Computer and System Sciences*, 47(2):250–286, 1993.

[18] Madhav V. Marathe, Heinz Breu, Harry B. Hunt III, S. S. Ravi, and Daniel J. Rosenkrantz. Simple heuristics for unit disk graphs. *Networks*, 25:59–68, 1995.

[19] S. Ramanathan. A unified framework and algorithm for channel assignment in wireless networks. *Wireless Networks*, 5(2):81–94, 1999.

[20] Subramanian Ramanathan and Errol L. Lloyd. Scheduling algorithms for multihop radio networks. *IEEE/ACM Transactions on Networking (TON)*, 1(2):166–177, 1993.

[21] Arunabha Sen and Mark L. Huson. A new model for scheduling packet radio networks. *Wireless Networks*, 3(1):71–82, 1997.

[22] Arunabha Sen and Ewa Melesinska. On approximation algorithms for radio network scheduling. In *Proceedings of the 35th Allerton Conference on Communication, Control and Computing*, pages 573–582, 1997.

[23] R. Sivakumar, B. Das, and V. Bharghavan. Spine routing in ad hoc networks. *ACM/Baltzer Cluster Computing Journal (special issue on Mobile Computing)*, 1998.

[24] I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks. In *Proceedings of IEEE Hawaii International Conference on System Sciences*, January 2001.

[25] Donald M. Topkis. All-to-all broadcast by flooding in communications networks. *IEEE Transactions on Computeres*, 38(9):1330–1333, 1989.

[26] P.-J. Wan, K. Alzoubi, and O. Frieder. Distributed construction of connnected dominating set in wireless ad hoc networks. In *IEEE INFOCOM*, 2002.

[27] J. Wu and H. L. Li. On calculating connected dominating set for efficient routing in ad hoc wireless networks. In *Proceedings of the Third ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pages 7–14, 1999.