Distributed Algorithms for Connected Domination in Wireless Networks

Rajiv Gandhi* Srinivasan Parthasarathy†

Abstract

We present fast distributed local control Connected Dominating Set (CDS) algorithms for wireless ad hoc networks. We present two randomized distributed algorithms, `CDSColor` and `CDSTop` which take into account the effect of wireless interference and the consequent loss of messages during the execution of the algorithm. These algorithms produce a CDS of constant size and constant stretch ratio with high probability, and converge in polylogarithmic running time. Specifically, algorithm `CDSColor` requires the nodes to know (estimates of) the maximum degree Δ and the size of the network n and converges in $O(\Delta \log^2 n)$ time. Algorithm `CDSTop` requires the nodes to know their three-hop topology and (an estimate of) the network size n and converges in $O(\log^2 n)$ time. To the best of our knowledge, these are the first distributed interference-aware CDS algorithms for wireless ad hoc networks which break the linear running-time barrier.

Keywords: Wireless Networks, Interference, Connected Dominating Set, Fast, Distributed Algorithms, Theory

1 Introduction

Ad hoc Networking has been a focus of intense research in recent years due to its tremendous potential in sensing, disaster relief, battle-field operations, community networking, etc. Ad hoc networks are composed of wireless mobile nodes and are characterized by a lack of a fixed communication infrastructure, frequent connectivity changes, strict bandwidth and power constraints, and wireless interference: characteristics which pose difficult challenges in the design of communication protocols for these networks. Several researchers have proposed the use of a virtual backbone in the network as an alternative to a fixed routing infrastructure [30, 10, 11]. Nodes in the virtual backbone act as a connected skeleton for the entire network and

*Department of Computer Science, Rutgers University, Camden, NJ 08102. E-mail: `rajivg@camden.rutgers.edu`. Part of this work was done when the author was a student at the University of Maryland and was supported by NSF Award CCR-9820965. Research also supported by Rutgers University's Research Council grant.

†Corresponding Author. Department of Computer Science, University of Maryland, College Park, MD 20742, USA. Phone: 240 485 8429. Fax: 301 405 6707. E-mail: `sri@cs.umd.edu`. Research supported by NSF Award CCR-0208005 and NSF ITR Award CNS-0426683.

frequently exchange local routing information (such as current traffic/mobility conditions, neighborhood information, etc.) so that other routing and group communication protocols can be implemented efficiently on top of the virtual backbone.

A Connected Dominating Set (CDS) is a natural candidate for virtual backbone infrastructure in ad hoc networks. A CDS is a connected subset of the network nodes such that any node in the network is either part of the CDS or has a neighbor in the CDS. For a virtual backbone to be effective, the underlying CDS must be small in size, have a low stretch (i.e., preserve the shortest paths in the original network), and must be *computable using fast distributed local control algorithms*. All currently known CDS algorithms in the literature can be essentially classified into three categories. **1)** Centralized algorithms which require all the network nodes to know the complete network topology [8, 14]. **2)** Fast localized distributed algorithms whose running time is constant ($O(1)$) or poly-logarithmic in the size of the network ($O(\log^k n)$, where k is a constant), but which do *not* take into account the effect of wireless interference [9, 32, 33]. **3)** Linear time localized distributed algorithms which handle the effect of wireless interference [2, 3, 4, 5, 6, 31]. These algorithms typically involve passing a token packet *sequentially* through each node in the network; the token contains information about the current CDS topology in the node's local neighborhood. Since the token is handled sequentially, there are no concurrent transmissions and no packet losses due to interference; however this imposes a linear running time on the algorithm.

Centralized CDS algorithms (from the first category) are hard to implement over an ad hoc network due to the difficulties involved in collecting and maintaining the entire network topology information at each node. Distributed algorithms which do not directly incorporate the effects of wireless interference also present difficult and open implementation challenges. In particular, it is unclear and open question if and how the underlying message transmissions in these algorithms could be scheduled such that the algorithm converges quickly (in constant or polylog time), *in the presence of message losses due to interference*. Linear-time distributed algorithms do not exploit the massive parallelism available in the ad hoc network. Further, linear-time algorithms are more susceptible to disruptions caused by network dynamics, since there is a high likelihood of changes in network connectivity before the termination of the distributed algorithm.

The above discussion naturally lead us to the following question: do there exist fast local control distributed algorithms for CDS construction in wireless ad hoc networks? In this work, we design two distributed CDS algorithms for wireless ad hoc networks, which answer this question in the affirmative. Specifically, the following are the main results presented in this paper.

1.1 Our Contributions

We present provably good fast interference-aware distributed algorithms for CDS construction in ad hoc networks. While several sub-linear distributed CDS algorithms exist in the literature, none of them explicitly incorporate the loss of messages due to wireless interference; Our algorithms are guaranteed to converge in polylogarithmic time and produce a valid CDS upon termination with high probability. To the best of our knowledge, *ours is the first work to present rigorous, fast converging, local control algorithmic techniques for CDS construction in wireless ad hoc networks which take into account message losses due to* *interference*, albeit with the assumption of synchronous network communication.

We present two distributed algorithms: a Distance-2 coloring based algorithm `CDSColor` and a neighborhood topology based algorithm `CDSTop`. Algorithm `CDSColor` requires each node to know the maximum degree Δ and the total number of network nodes n and runs in time $O(\Delta \log^2 n)$. Algorithm `CDSTop` requires each node to know their three-hop topology and runs in time $O(\log^2 n)$. Both these algorithms incorporate message loss due to collisions from interfering transmissions.

A basic underlying primitive for algorithm `CDSColor` is the distributed Distance-2 vertex coloring of nodes in the network. The Distance-2 coloring primitive arises naturally in many applications such as broadcast scheduling and channel assignment in wireless networks. 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 [29]. Our coloring algorithm runs in time $O(\Delta \log^2 n)$, where Δ is the maximum degree and n is the number of network nodes and uses $O(\Delta)$ colors for the D2-coloring; this is at most $O(1)$ times the number of colors used by an optimal algorithm. To the best of our knowledge, ours is the first distributed coloring algorithm known for the D2-coloring problem in wireless networks and could be of independent interest to other distributed scheduling applications as well.

The distributed CDS algorithms presented in this paper are optimized for constructing CDSs with certain powerful structural properties such as low size, low stretch and low degree, as introduced by Alzoubi [2]. We note that our algorithms and analysis only require that nodes know a good estimate of the values of the network parameters n and Δ 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. We note that while the assumption of network synchronization could be unrealistic for many ad hoc network scenarios, it is not too restrictive an assumption for networks enabled with a Global Positioning System (GPS). Yet another promising approach is to implement our algorithms on top of time synchronizers for wireless ad hoc networks (see Romer [28] for instance).

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 Sections 4 and 5, we present our distributed CDS algorithms `CDSColor` and `CDSTop` respectively. Section 6 contains concluding remarks and directions for future investigation.

2 Preliminaries

2.1 Network and Interference Model

We model the network using an undirected graph $G = (V, E)$: the nodes in V are embedded in the plane. Each node has a maximum transmission range and an edge $(u, v) \in E$ if 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., scaled to one unit). This model results in the network being modeled as a Unit Disk Graph (UDG).

Time is discrete and synchronized 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. We model interference using a simple receiver based model (**Rx-model**): if two or more neighbors of a node w transmit at the same time, w will be unable to receive any of those messages (but will be able to detect that a collision has occurred). In any time slot, a node can either receive a message, detect collision, or transmit a message but cannot do more than one of these.

*We use the UDG network model and the **Rx-model** for ease of exposition. Our techniques can easily be extended to a variety of other network and interference models that are frequently used to model ad hoc wireless networks.* For instance, in (r, s)-civilized graphs [18], any two nodes in the network are separated by a minimum distance of s and an edge between two nodes exist only if they are separated by a distance of at most r. The connectivity condition is only a necessary condition and not sufficient. In particular, lossy links or the presence of obstructions could be modeled in the above class of graphs by the absence of an edge between two nodes although they may have a distance $\leq r$ between them. Yet another frequently studied network model is that of a random wireless network, where nodes are placed randomly in a unit square and an edge exists between two nodes if they are at most a distance of r apart, with lossy links and obstructions being modeled by the absence of the corresponding edge in the graph. We note that all the above geometric network models share an important property of being growth restricted metrics which were introduced by Karger and Ruhl [17], which model several general and practical classes of network topologies; all the distributed algorithmic techniques presented in this paper extend to growth restricted metric graphs. Our techniques also extend to other interference models studied in the literature such as the **Tx-model** [34], the **Protocol model** [15], and the **Tx-Rx model** [21].

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

For the remainder of this paper, we let W denote a CDS. The following properties are often desirable in a CDS for efficient routing and scheduling.

(P1) Low Size: Let OPT be an MCDS in 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

Table 1 presents a comparison of the various distributed CDS algorithms known in the literature. Algorithms `CDSColor` and `CDSTop`, the two distributed algorithms presented in this work, are also featured in the table. The columns from left to right correspond to the following aspects of the algorithms respectively: the name of the algorithm, whether the algorithm explicitly takes into account the loss of messages due to wireless interference, the time complexity of the algorithm, the worst case approximation ratio of the size of the CDS, the worst case stretch of the CDS, whether the CDS algorithm requires the network to be synchronized, the graph classes to which the guarantees of the CDS algorithm apply, and the information required at each node during the execution of the algorithm. We remark that Dai *et al.* [9] present a constant upper bound on the size ratio of their distributed CDS algorithm for random Unit Disk Graphs (UDGs). The SPAN algorithm [7] is implemented over the 802.11 MAC layer protocol and hence the worst case convergence time for SPAN is not known (and assumed to be unbounded). We also note that the algorithm of Kuhn *et al.* [20] takes as input a value k, runs in time $O(k^2)$ and yields a dominating set of expected size $O(k\Delta^{\frac{2}{k}} \log \Delta)$ times the optimal. A preliminary version of the techniques developed in this paper also appears in Parthasarathy *et al.* [26]. As the table indicates, our distributed CDS algorithms are the first to incorporate wireless interference and converge in polylogarithmic time, albeit for synchronous networks.

Kuhn *et al.*[19] and Moscibroda *et al.*[25] propose fast interference-aware distributed algorithms for wireless networks; the proposed algorithms construct a dominating set and a maximal independent set and have time complexities, $O(\log^2 n)$ and $O(\frac{\log^3 n}{\log \log n})$ respectively. Significantly, both these papers assume an asynchronous quasi-UDG model for the ad hoc network and assume that the network nodes have no information about their local neighborhood and do not possess a reliable collision detection mechanism (i.e., nodes can not distinguish between collision and lack of transmission). However, unlike the techniques presented in this work, both these papers deal only with clustering the network and do *not* deal with obtaining a connected substructure of the network. We believe that the algorithmic techniques presented here and those of [19] and [25] are complementary to each other

Comparison of Distributed CDS Algorithms							
Algo	**Interference Aware**	**Time**	**Size**	**Stretch**	**Sync**	**Graph Class**	**Info**
Wan *et al.* [31]	yes	$O(n)$	$O(1)$	$O(n)$	no	UDG	D1
Alzoubi *et al.* [2, 1]	yes	$O(n)$	$O(1)$	$O(1)$	no	UDG	D1
Dai *et al.* [9]	no	$O(1)$	$O(n)^*$	$O(n)$	no	undirected	D2
Wu [33]	no	$O(1)$	$O(n)$	$O(n)$	no	directed	D2
Wu *et al.* [32]	no	$O(1)$	$O(n)$	$O(n)$	no	undirected	D2
SPAN [7]	yes	unbounded*	$O(1)$	$O(1)$	no	undirected	D3
Kuhn *et al.* [20]	no	$O(k^2)^*$	$O(k\Delta^{\frac{2}{k}} \log \Delta)$	$O(n)$	yes	undirected	Dk
Jia *et al.* [16]	no	$O(\log n \log \Delta)$	$O(\log \Delta)$	$O(n)$	yes	undirected	D1
Dubashi *et al.* [12]	no	$O(\log^2 n)$	$O(\log n)$	$O(\log n)$	yes	undirected	D1
`CDSColor`	yes	$O(\Delta \log^2 n)$	$O(1)$	$O(1)$	yes	UDG	n, Δ
`CDSTop`	yes	$O(\log^2 n)$	$O(1)$	$O(1)$	yes	UDG	D3,n

Table 1: Comparison of Distributed CDS Algorithms: Note how the fast converging CDS algorithms are typically not interference-aware and vice-versa. 'Graph Classes' UDG, undirected and directed represent Unit Disk Graphs, undirected and directed graphs respectively. 'Info' refers to the extent of local information (and global parameters) which each node is expected to know. * - See remarks in Section 3

and combining these approaches for developing faster CDS algorithms under more stringent network models merits future investigation.

D2-coloring is a problem with natural applications to conflict-free broadcast scheduling and channel assignment in wireless networks. In [29], 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 [29, 18, 27]. We present the first distributed implementation of D2-coloring for ad hoc wireless networks.

Luby[22, 23] proposed randomized distributed algorithms for vertex coloring and MIS construction in arbitrary undirected graphs. Our distributed algorithms are inspired by Luby's algorithms which were originally meant for a system of parallel processors. Our adaptations which are meant for wireless ad hoc networks are further constrained by the fact that messages can be lost due to collisions.

4 A centralized scheme

The distributed algorithms presented in this article are distributed implementations of a well-known centralized scheme for CDS construction. This scheme and its variants form the basic template for several distributed algorithms known in the literature as well as those developed in this work. Algorithm 1 takes as input a network $G = (V, E)$. Lines 2 to 7 compute a Maximal Independent Set (MIS) as follows: iteratively choose 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. Marathe *et al.*[24] first showed how this simple covering strategy results in $O(1)$ approximation for UDGs for covering problems such as minimum dominating sets. Lines 8 to 11 connect the nodes in MIS using *gateway* nodes. 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 these shortest paths which are not part of the MIS act as the gateway nodes; gateway nodes along with the nodes in MIS, constitute the CDS. The algorithm is illustrated in Figure 1.

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

$U = V$
$MIS = \phi$
while $U \neq \phi$ **do**
  Pick any $u \in U$
  $MIS = MIS \cup \{u\}$
  $U = U \setminus (\{u\} \cup N(u))$
end while
$CDS = MIS$
for all $\{u, v\} \subseteq MIS$ **do**
  if $D(u, v) \leq 3$ **then**
    $P =$ set of nodes in a shortest path from u to v
    $CDS = CDS \cup P$
  end if
end for
return CDS

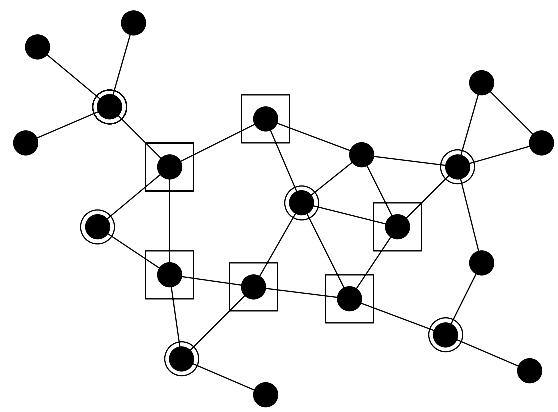

Figure 1: A CDS produced by the centralized algorithm: nodes enclosed in the circle are the MIS nodes; nodes within the squares are the *gateway* nodes; together they form the CDS.

Alzoubi *et al.* [2] proved that this scheme generates a CDS with constant stretch, constant degree and size within a constant factor of the optimal. Other variants of the this scheme trade off the size, degree and stretch guarantees: for instance,[31] builds a CDS with lower worst case degree and size ratio while sacrificing the constant stretch guarantee. As noted earlier, our distributed algorithms are distributed implementations of this scheme; however, our techniques can be easily modified to implement several other variants of this scheme as well, thus yielding the desired tradeoffs.

5 Distributed CDS Algorithms

In this section, we present our distributed CDS algorithms. Both the algorithms are randomized and can be parametrized to yield a valid CDS upon termination with very high probability. The basic contention resolution mechanism which underlies many stages of these distributed algorithms are as follows. Multiple neighboring nodes could contend for the channel at the same time, resulting in collisions and loss of messages at the intended recipients. However, the randomized transmission scheduling typically chooses transmitters and channel access probabilities such that each message has a constant probability of successful reception by its intended recipients. By repeating the randomized scheduling scheme a logarithmic number of times, messages are guaranteed to be successfully received by their intended recipients with high probability (w.h.p.)[1]. This is also the basis of the poly-logarithmic running times of our distributed algorithms. We now present the details of our distributed algorithms.

5.1 Algorithm `CDSColor`

We now present our first distributed CDS algorithm. Each node knows the maximum degree Δ, and the size of the network n. There are three stages in the algorithm. The first stage involves D2-coloring 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 $\{1, 2, \ldots c\}$. 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 color for themselves. Tentative color becomes permanent for a node, say v, if none of v's D2-neighbors have chosen the same tentative color as v. In this happens we say that v is successful. 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 becomes *awake* with probability $\frac{1}{2}$ and goes to sleep with the complementary probability. When a node goes to sleep it skips the current round. Each awake node u 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:

[1]In the rest of the paper, by w.h.p., we mean a probability value which is at least $1 - \frac{1}{n^\delta}$, where $\delta > 0$ is a constant which can be made arbitrarily large by appropriately choosing the parameters of the algorithm.

for e.g., if u is yet-uncolored 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 (recall that a frame contains c time slots), and broadcasts the TRIAL-REPORT message during this time.

SUCCESS: This phase consists of a single frame. At the beginning of this phase, every awake node u determines if the tentative color it chose during the TRIAL phase is a safe color or not. In our algorithm, u deems $color(u)$ to be safe if each TRIAL-REPORT message received by u contained the TRIAL message sent by u. In this case, $color(u)$ becomes the permanent color for u and 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 v does not choose the colors of its successful D2-neighbors in the future rounds.

This completes the description of a single round of the coloring stage. As mentioned earlier, the coloring stage consists of t such rounds, upon which the D2-coloring subroutine terminates.

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 received 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 CDS or not. All three stages are illustrated in Figure 2.

We now specify the values for the various parameters in the algorithm, which completes the description of the algorithm: $c = k_4\Delta$, $t = k_5 \log n$, and $r = k_6 \log n$, where k_4, k_5 and k_6 are suitable constants.

5.2 Analysis of Algorithm `CDSColor`

Theorem 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 in the plane. Let C be a disk in 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 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 3 *Let $G = (V, E)$ be a UDG with maximum degree Δ. 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 2 is at most $4k_0r^2$. Hence, the lemma follows. ■

Theorem 4 *All messages in the algorithm require at most $O(\Delta \log n)$ bits.*

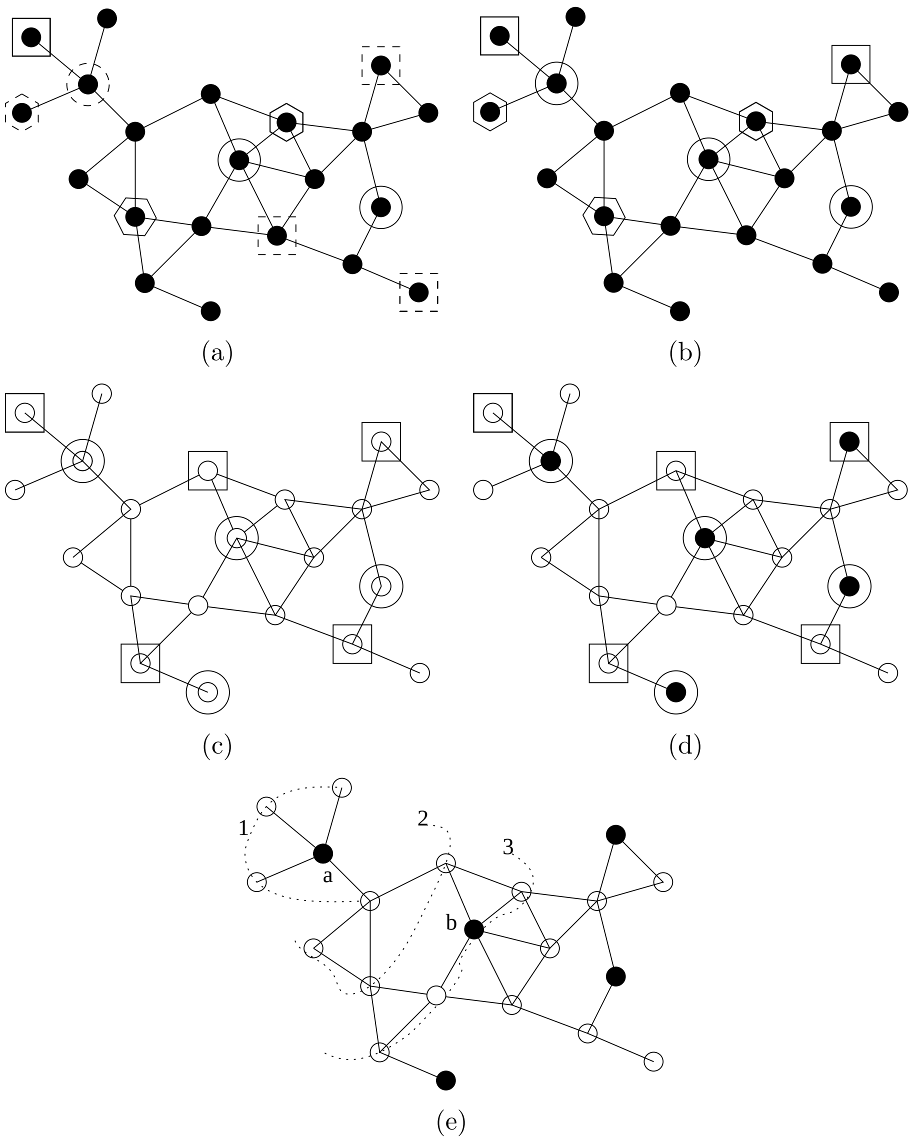

Figure 2: Algorithm `CDSColor`: sub figures (a) & (b) represent the TRIAL and SUCCESS phases of the D2-coloring stage; squares, hexagons and circles represent three distinct colors; solid enclosures represent permanent colors and dotted enclosures represent tentative colors; nodes that are not enclosed are not *awake*; note how two D2-neighbors enclosed in dotted squares are unsuccessful in sub figure (b). Sub figures (c) and (d) represent the MIS construction phase: nodes in (d) with black fillings are the MIS nodes; all nodes enclosed within circles join the MIS first; after this, only one node within a square can join the MIS; other colors are not shown for clarity. Sub figure (e) shows how the MIS nodes get connected; nodes along the dotted lines 1,2, and 3 know about the MIS node 'a' through PHASE-1, PHASE-2 and PHASE-3 messages; PHASES-4,5, and 6 are analogous.

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 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 concatenation of at most Δ messages of the previous types and hence require at most $O(\Delta \log n)$ bits. ■

Theorem 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 now prove the following lemmas which pertain to the correctness of the algorithm.

Lemma 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 $c = k_4\Delta$ is the initial size of the color-list for all nodes. Lemma 3 implies that the maximum number of D2-neighbors for any node is at most $16k_0(\Delta+1)$. Let the constant k_4 be chosen such that $k_4 > 32k_0$. 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 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 occurring 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 $c = k_4\Delta$, by letting $k_4 > 1$, 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,

$$\begin{aligned}
\Pr[v \text{ not receiving any TRIAL-REPORT from } u] &\leq \left(\frac{3}{4}\right)^{r} \\
&= \left(\frac{3}{4}\right)^{k_6 \log n} \\
&= \frac{1}{n^{\delta}}
\end{aligned}$$

Here, δ is a constant which can be made arbitrarily high by choosing an appropriate value of k_6. This completes the proof of the lemma. ■

Lemma 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 negligible low probability of $\frac{1}{n^{\delta}}$, where δ 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 7, both these events occur with the negligible low probability of $\frac{1}{n^\delta}$, where δ is a constant which can be made arbitrarily high. This completes the proof of the lemma. ■

Lemma 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 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 δ is a constant which can be made arbitrarily high. This completes the proof of the lemma. ■

Lemma 10 *Let u be a yet-uncolored node at the beginning of a particular round i. Let $S(u,i)$ denote the event that u was successfully colored during round i. Then,* $\Pr[\overline{S(u,i)}] \leq \frac{3}{4}$.

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, at least one of the the following two 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 $F(u)$.

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,

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

$$\leq \frac{1}{2|L_i(u)|}\sum_{x\in N}\frac{1}{2}$$
$$\leq \frac{|N|}{4|L_i(u)|}$$

By Lemma 6, $|L_i(u)| \geq |N| + 1$. Hence, the above probability is at most $\frac{1}{4}$. Hence, we have

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

which completes the proof of the lemma. ■

Lemma 11 *When D2-coloring algorithm terminates, w.h.p. all nodes are successfully colored.*

Proof By Lemma 10, the probability of a particular node u remaining yet-uncolored after the algorithm terminates is at most $\left(\frac{3}{4}\right)^t = \left(\frac{3}{4}\right)^{k_5 \log n} = \frac{1}{n^\delta}$. 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, δ is a constant which can made arbitrarily high by choosing the appropriate value of k_5. This concludes the proof of the lemma. ■

Theorem 12 *The first stage computes a valid D2-coloring w.h.p.*

Proof Lemmas 9 and 11 together yield this theorem. ■

Theorem 13 *The second and third stages compute a valid MIS and CDS 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 12, the first stage computes a valid D2-coloring w.h.p. Hence this theorem follows. ■

5.3 Algorithm `CDSTop`

We now describe our distributed CDS algorithm based on neighborhood topology information. Each node is assumed to have knowledge of its D3-topology, i.e., nodes in its three hop neighborhood and the edges between these nodes. The algorithm comprises of two stages. An MIS is constructed in the first stage and it is connected in the second stage. Both the first and the second stage of the algorithm involve the use of the collision-free broadcasting algorithm of Gandhi *et al.* [13], to broadcast messages to D2 and D3-neighborhoods respectively: the broadcast subroutine utilizes the D2 and D3-topology information to create a collision-free broadcast schedule. We now 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 the yet-unsuccessful ones continue their attempts 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 involve 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 experiences 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 α sub-frames of length c. During *every* frame of this phase, each node in MIS_i, chooses independently at random, one of the α sub-frames. During this sub-frame, it broadcasts a PREPARE message using the algorithm in [13] to its D2-neighbors. The length of the sub-frame, c is the number of time slots required by [13] 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. Since it knows its D3 (and hence D2) topology, it can easily compute the value $N_{i+1}(u)$. Note that, since the message is being broadcast to the two-hop neighborhood, the broadcast algorithm guarantees that c is at most a constant [13].

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.

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 α' sub-frames of length c'. During *every* frame of this phase, each node $u \in MIS$ node selects independently at random, one of the α' sub-frames. During this sub-frame, u broadcasts a HELLO message using the algorithm in [13] 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. In general, intermediate nodes which are not part of the MIS may join the CDS, since they could be a part of the shortest path between two MIS nodes.

We complete the description of our topology based distributed CDS algorithm by specifying the values of the various parameters involved: we let $p = k_7 \log n$ and $t = k_8 \log n$. We let α and α' be the maximum number of MIS nodes in the D2 and D3-neighborhoods of any node respectively: these values are fixed constants [13]. Let c and c' be the maximum number of time slots required by the broadcast algorithm [13] to broadcast a message to the D2 and D3-neighborhoods of a node respectively: both these values are fixed constants as well [13]. Due to lack of space, we state only the main claim pertaining to the performance analysis of our algorithm here. The detailed proof of this claim involves several subtle probabilistic and geometric arguments; the reader is referred to Figure 3 for an illustration of the algorithm.

5.4 Analysis of Algorithm `CDSTop`

Theorem 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 α, α', c, and c' are constants, the running time is $O(tp) = O(\log^2 n)$. ■

Theorem 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 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.*

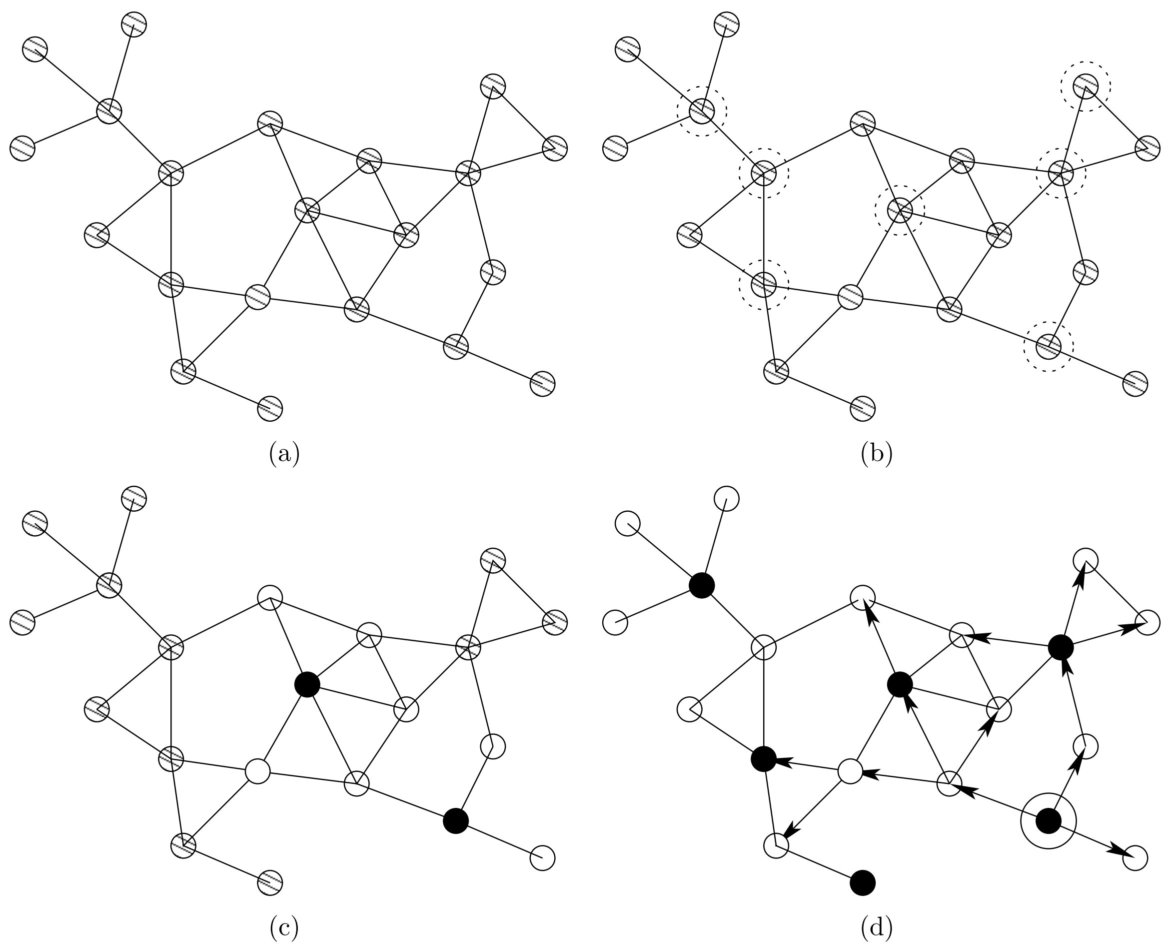

Figure 3: Algorithm `CDSTop`: subfigures (a), (b), & (c) show the MIS construction phase: nodes filled with the pattern are *unsure* nodes; nodes within the dotted circles in (b) are the *candidate* nodes; in subfigure (c), nodes with black fillings are *in* the MIS and nodes with white fillings are *out* of the MIS. Subfigure (d) shows the final MIS; note the 3-hop broadcast tree rooted at the encircled MIS node; such 3-hop broadcast trees are employed by MIS nodes to inform and connect to other MIS nodes within the 3-hop neighborhood.

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.

$$\begin{aligned} \Pr[\bigwedge_{j=1}^{p} A(u, v, j)] &\leq \frac{1}{2^p} \\ &= \frac{1}{2^{k_7 \log n}} \\ &= \frac{1}{n^\delta} \end{aligned}$$

Here, δ can be made arbitrarily high by choosing an appropriate value of k_7. ■

Lemma 17 *Consider a round i and a node $u \in MIS_i$. Consider a D2-neighbor v of u. v receives (at least one of) the PREPARE message transmitted by u collision-free w.h.p. during the **PREPARE** phase of round i.*

Proof Recall that the **PREPARE** phase is divided into p frames, each of which is divided into α subframes, where the constant α is the maximum number of MIS nodes in the D3-neighborhood of any node. Let $F(u, j)$ denote the sub-frame of frame j during which u broadcast its PREPARE message to its D2-neighborhood. Let S denote the D3-neighborhood of v. If v does not receive u's PREPARE message during the sub-frame $F(u, j)$ of frame j, then there exists $w \in S \setminus \{u\}$ such that the event $F(w, j) = F(u, j)$ occurred.

$$\begin{aligned} \Pr[\exists w \in S \setminus \{u\} F(w, j) = F(u, j)] &= 1 - \Pr[\forall w \in S \setminus \{u\} F(w, j) \neq F(u, j)] \\ &= 1 - \Pi_{w \in S \setminus \{u\}} \Pr[F(w, j) \neq F(u, j)] \\ &= 1 - \Pi_{w \in S \setminus \{u\}} (1 - \frac{1}{\alpha}) \\ &= 1 - \left(1 - \frac{1}{\alpha}\right)^{|S|-1} \\ &= 1 - \left(1 - \frac{1}{\alpha}\right)^{\alpha-1} \\ &\leq 1 - \frac{1}{e} \end{aligned}$$

Let $B(u, v, j)$ denote the bad event that v does not receive u's prepare message during frame j. The above inequalities imply $\Pr[B(u, v, j)] \leq 1 - \frac{1}{e}$. Node v will not receive any of the PREPARE messages transmitted by u if the event $\bigwedge_{j=1}^{p} B(u, v, j)$ occurs.

$$\Pr[\bigwedge_{j=1}^{p} B(u, v, j)] = \Pi_{j=1}^{p} \Pr[A(u, v, j)]$$

$$
\begin{aligned}
&\le \left(1 - \frac{1}{e}\right)^{p} \\
&\le \left(1 - \frac{1}{e}\right)^{k_7 \log n} \\
&\le \frac{1}{n^{\delta}}
\end{aligned}
$$

Here, δ is a constant which can be made arbitrarily large by choosing an appropriate value of k_7. This completes the proof of the lemma. ■

Lemma 18 *Consider a node $u \in MIS_i$ and a D3-neighbor v of u. Node v receives (at least one of) the HELLO (CONNECT) messages transmitted by u collision-free w.h.p. during the* ***HELLO (CONNECT)*** *phase of the second stage.*

Proof The proof of this lemma is identical to the proof of lemma 17, except for the following differences: the PREPARE message is broadcast to D2-neighborhood of a node, whereas the HELLO (CONNECT) message is broadcast to the D3-neighborhood. The constants α and c in the proof of lemma 17 are replaced by the constants α' and c' in this proof. ■

Lemma 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 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 occurring 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 δ (and hence β) can be made arbitrarily high. Similarly, by lemmas 17 and 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 β can be made arbitrarily high. Hence, none of the three bad events occur w.h.p. ■

Lemma 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 19 occur. These events occur with negligible probability. Hence the lemma follows. ■

Lemma 21 *If MIS is a maximal independent set, then the second stage computes a valid CDS w.h.p.*

Proof Assume that the first stage computes an MIS which is a valid maximal independent set. Then the CDS computed in the second stage will not be valid, only if the third bad event in Lemma 19 occurs. Lemma 19 states that this event does not occur during the second stage, w.h.p. Hence the lemma follows. ■

We now prove that the MIS is also maximal, i.e., all nodes are either *in* or *out* of the MIS after the first stage. 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 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 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 $v \in V_i(H)$,* $\Pr[X_i(H) = 1 | X_i(H) \geq 1] \geq \frac{1}{2}$.

Proof Let $Y_i(H)$ be the random variable which denotes the number of *candidates* in $V_(H) \setminus \{v\}$. Clearly, $X_i(H) = Y_i(H) + 1$.

$$\begin{aligned}
\Pr[X_i(H) \geq 2 | C(v,i)] &= \Pr[Y_i(H) \geq 1 | C(v,i)] \\
&= \Pr[Y_i(H) > 0 | C(v,i)]
\end{aligned}$$

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

■

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 24 $\Pr[u \in MIS_i|C(u,i)] \geq \beta$*, where* $\beta > 0$ *is a constant.*

Proof

$$
\begin{aligned}
\Pr[u \in MIS_i|C(u,i)] &\geq \Pr[(X_i(H)=1) \bigwedge (\forall (s \in S\setminus\{H\})(X_i(s)=0))|C(u,i)] \\
&\geq \Pr[(X_i(H)=1)|C(u,i)] \Pr[(\forall (s \in S\setminus\{H\})(X_i(s)=0))|C(u,i)] \\
&\geq \frac{1}{2}\Pi_{s\in S\{H\}} \Pr[X_i(s)=0|C(u,i)] \\
&\geq \frac{1}{2}\left(\frac{1}{2^{|S|-1}}\right) \\
&\geq \frac{1}{2^{|S|}} \\
&= \beta
\end{aligned}
$$

Claim 2 implies that $|S|$ is $O(1)$. Hence, $\beta > 0$ 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 2 implies that the maximum degree of any node in F is at most a constant ψ. 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 prove the following claims.

Claim 25 *Let* H *be a vertex in* $\mathcal{F}$ *and let* $Children(H)$ *be the children of* H *in* $\mathcal{F}$*. Then,* $|V_i(H)| \geq 2(\Sigma_{H'\in Children(H)}|V_i(H')|)$*.*

Proof The number of vertices adjacent to H in $\mathcal{F}$ (and F) is at most ψ. If (H, H') is an edge in $\mathcal{F}$, then $|V(H)_i| \geq 2\psi|V_i(H')|$. Hence, $\frac{|V_i(H)|}{\psi} \geq 2|V_i(H')|$. Since, there are at most ψ children of H, the claim follows. ■

Claim 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 $\mathcal{T}$.*

Proof Let $Height(H)$ denote the height of H in the tree $\mathcal{T}$. The leaf nodes have a height of 0. Their parents have a height of 1 and so on. The proof is by induction on $Height(H)$.
Base Case: $Height(H) = 0$. In this case, H is a leaf node and has no descendents. Hence the claim holds.
Induction Hypothesis: Let the claim hold for all nodes such that their height is $\leq h$.
Induction Step: We now prove the claim for a node H such that $Height(H) = h + 1$. For any $H_1 \in Children(H)$, by the induction hypothesis, $|V_i(H_1)| \geq \Sigma_{H_2 \in Desc(H_1)} |V_i(H_2)|$. Hence,

$$\begin{aligned} |V(H) \bigcap V_i| &\geq 2(\Sigma_{H_1 \in Children(H)} |V_i(H_1)|) \\ &\geq \Sigma_{H_1 \in Children(H)} 2|V_i(H_1)| \\ &\geq \Sigma_{H_1 \in Children(H)} (|V_i(H_1)| + \Sigma_{H_2 \in Desc(H_1)} |V_i(H_2)|) \\ &= \Sigma_{H' \in Desc(H)} |V_i(H')| \end{aligned}$$

This completes the proof of the claim. ■

Claim 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.*

Proof Since H is the root of $\mathcal{T}$, it does not have any in-edge incident upon it. Hence, for all $u \in V_i(H)$, $\Pr[C(u,i)] \geq \frac{1}{4\psi^2 |V_i(H)|}$. Hence,

$$\begin{aligned} \Pr[|X_i(H)| > 0] &\geq \Sigma_{u \in V_i(H)} \Pr[C(u,i)] \\ &\geq \Sigma_{u \in V_i(H)} \frac{1}{4\psi^2 |V_i(H)|} \\ &\geq \frac{1}{4\psi^2} = \gamma \end{aligned}$$

■

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 28 $\mathbf{E}[|Z_i|] \geq \kappa |V_i|$, *where $\kappa > 0$ is a constant.*

Proof Let $\mathcal{T}$ be any tree in the forest $\mathcal{F}$. For any $H' \in \mathcal{T}$, let $Z_i(H')$ be the set of nodes in $V_i(H')$ which are *candidates* for MIS_i or which have a neighbor who is a *candidate* for MIS_i. We now show that $\Sigma_{H' in \mathcal{T}} |Z_i(H')| \geq \gamma \Sigma_{H' in \mathcal{T}} |V_i(H')|$, where $\gamma > 0$ is a constant. If this holds for every tree in the forest, then clearly the lemma holds. We now prove the above inequality for $\mathcal{T}$. Let H be the root of $\mathcal{T}$. Since there is no in-edge (H', H) incident upon H, the following holds:

$$\begin{aligned}
\Sigma_{H' \in \mathcal{T}} \mathbf{E}[|Z_i(H')|] &\geq \mathbf{E}[|Z_i(H)|] \\
&\geq |V_i(H)| \Pr[|X_i(u) > 0] \\
&\geq \frac{1}{2} \beta \Sigma_{H' \in \mathcal{T}} |V_i(H')| \gamma \\
&\geq \kappa(\Sigma_{H' \in \mathcal{T}} |V_i(H')|)
\end{aligned}$$

This concludes the proof of the lemma. ■

Let ϵ be the minimum probability of a fixed *candidate* in a round i joining MIS_i. Lemma 24 ensures that $\epsilon > 0$ is at least 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 29 $\mathbf{E}[|succ_i|] \geq \epsilon|V_i|$, *where* $\epsilon > 0$ *is a constant. Hence, for all* $i > 0$, $\mathbf{E}[|V_i|] \leq (1 - \epsilon)\mathbf{E}[|V_{i-1}|]$.

Proof

$$\begin{aligned}
\mathbf{E}[|succ_i|] &\geq \mathbf{E}[|Z_i|] \min_{u \in X_i} \Pr[u \in MIS_i] \\
&\geq \beta \mathbf{E}[|Z_i| \\
&\geq \beta\kappa|V_i| \\
&\geq \epsilon|V_i|
\end{aligned}$$

■

Theorem 30 *The expected number of messages transmitted during the algorithm is* $O(n \log n)$.

Proof During the second stage, each MIS node broadcasts at most $2p$ messages to its D3-neighborhood. Each of these broadcasts involve $O(1)$ transmissions [13]. 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 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 29,

$$\begin{aligned}\Pr[|V_{p+1}| > 0] &\leq \mathbf{E}[|V_{p+1}|] \\ &\leq |V_1|(1-\epsilon)^p \\ &\leq n(1-\epsilon)^{k_7 \log n} \\ &\leq \frac{n}{n^{\delta+1}} \\ &\leq \frac{1}{n^{\delta}}\end{aligned}$$

where $\delta > 0$ is a constant which can be made arbitrarily large by choosing an appropriate value of k_7. Hence the lemma follows. ■

Theorem 32 *The MIS computed by the first stage is valid w.h.p.*

Proof By Lemma 19, MIS is an independent set w.h.p. Lemma 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 33 *The CDS computed at the end of the second stage is valid with high probability.*

Proof Theorem 32 implies that the MIS is valid w.h.p. Lemma 21 implies that if the MIS is valid, then the CDS is valid w.h.p. Hence the theorem follows. ■

6 Conclusion and Future Work

We presented fast distributed CDS algorithms for ad hoc wireless networks, which compute a CDS in poly-logarithmic time, even after taking into account, the effects of wireless interference. These are the first such interference-aware distributed virtual backbone algorithms which provably break the linear time barrier. Interesting future extensions include designing provably good fast distributed algorithms which incorporate the effect of node mobility and relax the assumptions of network synchronization. Both these extensions present fundamental analytical challenges and has tremendous practical significance for large scale ad hoc wireless networks.

Acknowledgments. We would like to thank V.S. Anil Kumar, Madhav Marathe and Aravind Srinivasan for several useful discussions.

References

[1] K. Alzoubi, P.-J. Wan, and O. Frieder. Maximal independent set, weakly connected dominating set, and induced spanners for mobile ad hoc networks. *International Journal of foundations of Computer Science (IJFCS), Special Issue on Wireless*, 2003.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

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

[7] 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.

[8] 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.

[9] Fei Dai and Jie Wu. An extended localized algorithm for connected dominating set formation in ad hoc wireless networks. *IEEE Trans. on Parallel and Distributed Systems*, 15(10):908–920, 2004.

[10] 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.

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

[12] 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.

[13] 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.

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

[15] Piyush Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, March 2000.

[16] Lujun Jia, Rajmohan Rajaraman, and Torsten Suel. An efficient distributed algorithm for constructing small dominating sets. *Distrib. Comput.*, 15(4):193–205, 2002.

[17] David R. Karger and Matthias Ruhl. Finding nearest neighbors in growth-restricted metrics. In *STOC '02: Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 741–750. ACM Press, 2002.

[18] 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.

[19] Fabian Kuhn, Thomas Moscibroda, and Roger Wattenhofer. Initializing newly deployed ad hoc and sensor networks. In *10th Annual International Conference on Mobile Computing and Networking (MOBICOM)*, 2004.

[20] Fabian Kuhn and Roger Wattenhofer. Constant-time distributed dominating set approximation. In *Proceedings of the twenty-second annual symposium on Principles of distributed computing*, pages 25–32. ACM Press, 2003.

[21] V. S. Anil Kumar, Madhav V. Marathe, Srinivasan Parthasarathy, and Aravind Srinivasan. End-to-end packet scheduling in ad hoc networks. In *ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 1014–1023, 2004.

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

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

[24] 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.

[25] Thomas Moscibroda and Roger Wattenhofer. Efficient computation of maximal independent sets in unstructured multi-hop radio networks. In *1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, 2004.

[26] Srinivasan Parthasarathy and Rajiv Gandhi. Distributed algorithms for coloring and connected domination in wireless ad hoc networks. In *Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, 2004.

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

[28] Kay Romer. Time synchronization in ad hoc networks. In *MobiHoc '01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, pages 173–182. ACM Press, 2001.

[29] 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.

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

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

[32] Jie Wu. Dominating-set-based routing in ad hoc wireless networks. *Handbook of wireless networks and mobile computing*, pages 425–450, 2002.

[33] Jie Wu. Extended dominating-set-based routing in ad hoc wireless networks with unidirectional links. *IEEE Trans. on Parallel and Distributed Systems*, 13(9):866–991, 2002.

[34] Su Yi, Yong Pei, and Shivkumar Kalyanaraman. On the capacity improvement of ad hoc wireless networks using directional antennas. In *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 108–116. ACM Press, 2003.