Distributed Algorithms for Coloring and Domination in Wireless Ad Hoc Networks

Srinivasan Parthasarathy[1] and Rajiv Gandhi[2]

[1] Department of Computer Science, University of Maryland, College Park, MD 20742. Research supported by NSF Award CCR-0208005. E-mail: sri@cs.umd.edu.
[2] 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.

Abstract. We present fast distributed algorithms for coloring and (connected) dominating set construction in wireless ad hoc networks. We present our algorithms in the context of Unit Disk Graphs which are known to realistically model wireless networks. Our distributed algorithms take into account the loss of messages due to contention from simultaneous interfering transmissions in the wireless medium.
We present randomized distributed algorithms for (conflict-free) Distance-2 coloring, dominating set construction, and connected dominating set construction in Unit Disk Graphs. The coloring algorithm has a time complexity of $O(\Delta \log^2 n)$ and is guaranteed to use at most $O(1)$ times the number of colors required by the optimal algorithm. We present two distributed algorithms for constructing the (connected) dominating set; the former runs in time $O(\Delta \log^2 n)$ and the latter runs in time $O(\log^2 n)$. The two algorithms differ in the amount of local topology information available to the network nodes.
Our algorithms are geared at 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.

1 Introduction

Wireless 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 distributed, wireless and self-configuring nature of ad hoc networks render them useful for several applications such as mobile battlefields, disaster relief, sensing and monitoring. However, the lack of a fixed communication infrastructure introduces several challenging and interesting research issues in the design of communication protocols for these networks. Any communication protocol for ad hoc networks should also contend with the issue of interference in the wireless medium. When two or more nodes transmit a message to a common neighbor at the same time, the common node will not receive any of these messages. In such a case, we say that a collision has occurred at the common node.

Coloring and connected domination are two fundamental primitives with several applications in the wireless context. In wireless networks, we seek a conflict-free coloring of the nodes such that two nodes which belong to the same color class may transmit simultaneously without resulting in collisions. Clearly, such a coloring has natural applications to collision-free wireless scheduling. In order to overcome the lack of a fixed routing infrastructure, several researchers have also proposed construction of a *virtual backbone* in ad hoc 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 [9, 8, 20].

Both coloring and (connected) dominating set construction are classical problems which have received tremendous attention in the literature. In general, all existing distributed algorithms for these problems can be classified into two categories. The first category of algorithms are fast sub-linear time algorithms which do *not* consider message losses due to collisions. Further, these all algorithms model the network as an arbitrary undirected graph; both these assumptions render them unsuitable for wireless ad hoc networks. The second category of algorithms are (slower) 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. A linear time algorithm does not exploit the massive parallelism available in the ad hoc network and is unsuitable for dynamic network conditions displayed by ad hoc networks.

In this work, we focus on developing fast distributed algorithms for coloring and (connected) dominating set construction in wireless ad hoc networks. Specifically, we view the following as the main contributions of this work.

1.1 Our Contributions

– **Incorporating Wireless interference:** We present distributed algorithms for conflict-free coloring, dominating set construction and connected dominating sets in the context of wireless networks. While several distributed algorithms exist for coloring and domination in arbitrary graphs, we use Unit

Disk Graphs which realistically model wireless networks. Further, our algorithms handle wireless interference; we take into account the loss of messages at a node due to collisions from simultaneous neighboring transmissions. We are not aware of any work which study these problems under message losses due to wireless collision.

- **Distributed Coloring** We present a distributed conflict-free (D2) coloring of nodes in the network. This 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 [19]. Our 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.
- **Distributed (Connected) Dominating Set:** We present distributed algorithms for dominating set and connected dominating set construction where require knowledge of only local topology and global network parameters such as size and the maximum degree. We present two algorithms: a D2-coloring based algorithm and a broadcast based algorithm which utilizes the work of Gandhi *et al.* [10]. The coloring based algorithm requires each node to know the maximum degree Δ and the total number of network nodes n and runs in time $O(\Delta \log^2 n)$. The broadcast based algorithm requires each node to know their three-hop topology and runs in time $O(\log^2 n)$. All these algorithms incorporate message losses due to collisions from interfering transmissions.
- **Wireless Routing Applications:** The distributed CDS algorithms presented in this paper are geared at 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 wireless 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.

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. Due to lack of space, we omit the proofs of all the claims presented in this paper. All the proofs appear in the full version of this work. [3]

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 $(u, v) \in E$ 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., 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 easily extend to the so called **protocol model** [11] 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 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 $M' \supseteq M$ 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.

[3] Available at http://www.cs.umd.edu/∼sri/distcoldom.ps

(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

Coloring, dominating set construction and connected domination are classical problems which have been extensively studied in the literature. However, we are not aware of any distributed algorithms for these problems which incorporate the geometry and transmission characteristics of wireless networks. To the best of our knowledge, we are the first to study these problems for realistic multi-hop wireless network models (Unit Disk Graphs) and incorporate loss of messages due to collisions from interfering transmissions. In [19], 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 [19, 12, 18]. It was shown in [7] that computing an MCDS is NP-hard even for UDGs. Cheng *et al.*[6] propose a centralized polynomial time approximation scheme (PTAS) for approximating MCDS in UDGs. Several distributed approximation algorithms exist for computing MCDS in UDGs [21, 16, 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. 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 $O(n \log n)$). While the distributed algorithm presented in [1] holds for both synchronous and asynchronous models of communication, we restrict our focus only to the synchronous communication model and leverage in the design of our distributed algorithms.

4 Distributed D2-Coloring

In this section, we present our distributed D2-coloring algorithm for unit disk graphs. Our algorithm is modeled after Luby's distributed graph coloring algorithm [15]. The key technical difficulty in our algorithm as opposed to Luby's algorithm, lies in the the fact that simultaneous transmissions from neighboring nodes could result in collisions and hence loss of messages at a particular node. We handle this by probabilistic retransmission of the messages, and ensure that all messages are eventually received by their intended recepients with high probability. Further, while Luby's distributed coloring algorithm was a *D1-coloring of arbitrary graphs*, our algorithm is intended for *D2-coloring of unit disk graphs*. This allows us to exploit the geometric properties of UDGs to D2-color it using $O(\Delta)$ colors; this yields a $O(1)$ approximation for the number of colors.

Our algorithm 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*. As in Luby's algorithm [15], our algorithm also 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 if 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 algorithm; the algorithm consists of t such rounds. We show that for an appropriate choice of parameters, our algorithm yields a $O(1)$-approximate D2-coloring for UDGs with high probability in $O(\Delta \log^2 n)$ time. Specifically, let the parameters 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. The following theorem holds.

Theorem 1. *The distributed D2-coloring algorithm computes a valid D2-coloring using $O(\Delta)$ colors in $O(\Delta \log^2 n)$ running time w.h.p. The number of colors used is at most $O(1)$ times the optimal coloring. All messages in the algorithm require at most $O(\Delta \log n)$ bits. The total number of messages transmitted by the algorithm is at most $O(n \log^2 n)$.*

5 Distributed Dominating Set Construction

In this section, we present our distributed dominating set algorithms for unit disk graphs. We note that any Maximal Independent Set (MIS) is also a dominating set in an undirected graph. Further, in the case of UDGs, it is well known that the number of nodes in *any* Maximal Independent Set (MIS) is at most five times the number of nodes in the minimum dominating set. Hence, a distributed MIS algorithm also yields a 5-approximate dominating set in UDGs. Henceforth, we focus on distributed MIS construction in UDGs.

5.1 D2-coloring based MIS Algorithm

We now present a simple D2-coloring based distributed MIS algorithm. Observe that if we have a D2-coloring of the nodes using c colors, we can build an MIS iteratively in c time slots as follows: 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. For the same reason, none of the messages are lost due to collisions. Clearly, this stage requires exactly c time steps. Since the distributed D2-coloring algorithm of Section 4 colors the UDG using $O(\Delta)$ colors w.h.p. in $O(\Delta \log^2 n)$ time, we also have a distributed MIS algorithm which terminates correctly w.h.p. in $O(\Delta \log^2 n)$ time.

Theorem 2. *The D2-coloring based distributed algorithm constructs an MIS in $O(\Delta \log^2 n)$ time w.h.p and the total number of messages transmitted during the algorithm is $O(n \log^2 n)$.*

5.2 Broadcast based MIS construction

We now present our broadcast based distributed MIS algorithm, which makes use of knowledge of the Distance-2 topology, and constructs an MIS in $O(\log^2 n)$ time. Specifically, we assume that each node knows its D2-neighborhood and the edges between these nodes. As in Luby's distributed MIS algorithm [14], our algorithm also 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 (except for forwarding messages), while remaining nodes continue their attempts to be successful in the future rounds. The MIS construction terminates after t such rounds.

During the algorithm, each node u maintains a status variable which is defined as follows: status(u)=*in* if 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 algorithm: **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 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 [10] to its D2-neighbors. The length of the sub-frame, c is the number of time steps required by [10] 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 $N_{i+1}(u)$.

The algorithm terminates after t such rounds. The theorem below claims that for an appropriate choice of parameters, the algorithm yields an MIS with high probability in time $O(\log^2 n)$. The analysis of this theorem involves a tricky charging argument which heavily relies on the geometry of UDGs.

Theorem 3. *The broadcast based distributed algorithm computes an MIS with high probability in $O(\log^2 n)$ time. Each message is at most $O(\log n)$ bits in length and the expected number of messages transmitted is $O(n \log n)$.*

6 Distributed Connected Domination

In this section, we present the results for our distributed connected dominating set algorithms for UDGs. Alzoubi [1] presented a centralized algorithm for constructing a CDS with a stretch of $O(1)$, size which is at most $O(1)$ times that of the minimum CDS, and has $O(1)$ degree. Henceforth, we will call a CDS with these properties as Well Connected Dominating Set (or WCDS). Alzoubi also presented a distributed implementation of his centralized algorithm which runs in linear time. The basic idea behind the centralized algorithm is as follows: we first compute an MIS by iteratively choosing vertices which are currently not in MIS and which do not currently have a neighbor in MIS. Since the input graph is an undirected graph, any maximal independent set is also a dominating set.

Connectivity is handled as an orthogonal component as follows: 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 CDS W.

We present two distributed implementations of this approach. Due to lack of space, the details of these implementations are presented in the full version. We note that in both the algorithms, the basic idea is for each node in the MIS to broadcast a message to its D3-neighborhood. After this step, each node in the MIS connects itself to every other node in the MIS which is at most three hops away, through a shortest path. The two implementations differ in how the MIS is computed and how this broadcasting is achieved. The first implementation uses the D2-coloring based scheme. The broadcasting is easily achieved in a collision-free manner since the D2-coloring also yields a natural collision-free schedule. The running time for this algorithm is $O(\Delta \log^2 n)$ and is dominated by the D2-coloring step. In the second implementation, the MIS is constructed via the broadcast algorithm discussed in Section 5.2. Here the broadcasting is achieved using the algorithm in [10]. This implementation has a running time of $O(\log^2 n)$ and is dominated by the MIS construction. As discussed in Section 5, these algorithms differ in extent of local topology information available to each node in the network.

7 Network-wide Broadcasting

We now present our results pertaining to our broadcast algorithm. Due to lack of space, the details of the algorithm are presented in the full version. The basic idea behind the broadcast algorithm is to first construct a WCDS W, and obtain a valid D2-coloring of the WCDS. 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 using k colors. Let time be divided into frames of length k. Every node in the WCDS, retransmits a message after receiving it, in the first time slot in the following frame which corresponds to its own color. If there are multiple messages to be transmitted, the one with the lowest ID is chosen for transmission. 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. We analyze the behavior of our broadcast algorithm under the following packet injection model.

Theorem 4. *The broadcast algorithm supports an long term rate of message generation, which is within $O(1)$ factor of the optimal rate. Further, the latency experienced by any message is at most $O(1)$ times the optimal latency for this message. 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.*

8 Unicast Routing

In this section, we show that a 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) [17, 18, 13]. We show that any routing algorithm could be modified to operate over a WCDS such that, 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 $\mathcal{P} = \{p_1, \ldots p_n\}$ 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 *congestion* 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 [13].

Claim. 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, $|OPT| = \Theta(c + d)$.

The following theorem holds.

Theorem 5. *There exists a set of paths $\mathcal{P}'$ such that each path in $\mathcal{P}$ can be replaced by an alternate path in $\mathcal{P}'$. Further, these paths are such that all their internal nodes are from the WCDS and congestion c' and dilation d' of the path system $\mathcal{P}'$ are such that $c' + d' = \Theta(c + d)$.*

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

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 non-trivial 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. 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.
7. B. Clark, C. Colbourn, and D. Johnson. Unit disk graphs. *Discrete Mathematics*, 86:165–177, 1990.
8. 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.
9. Bevan Das and Vaduvur Bharghavan. Routing in ad-hoc networks using minimum connected dominating sets. In *ICC (1)*, pages 376–380, 1997.
10. 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.
11. Piyush Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, March 2000.
12. 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.
13. 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.
14. Michael Luby. A simple parallel algorithm for the maximal independent set problem. *SIAM Journal on Computing*, 15(4):1036–1053, November 1986.
15. Michael Luby. Removing randomness in parallel computation without a processor penalty. *Journal of Computer and System Sciences*, 47(2):250–286, 1993.
16. 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.
17. S. Ramanathan. A unified framework and algorithm for channel assignment in wireless networks. *Wireless Networks*, 5(2):81–94, 1999.
18. Subramanian Ramanathan and Errol L. Lloyd. Scheduling algorithms for multihop radio networks. *IEEE/ACM Transactions on Networking (TON)*, 1(2):166–177, 1993.
19. 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.
20. R. Sivakumar, B. Das, and V. Bharghavan. Spine routing in ad hoc networks. *ACM/Baltzer Cluster Computing Journal (special issue on Mobile Computing)*, 1998.
21. P.-J. Wan, K. Alzoubi, and O. Frieder. Distributed construction of connnected dominating set in wireless ad hoc networks. In *IEEE INFOCOM*, 2002.