The Distance-2 Matching Problem and its Relationship to the MAC-layer Capacity of Ad hoc Wireless Networks

Hari Balakrishnan [3], Christopher L. Barrett [1], V.S. Anil Kumar [1], Madhav V. Marathe [1], and Shripad Thite [2]

***Abstract*— We consider the problem of determining the maximum capacity of the MAC layer in wireless ad hoc networks. Due to spatial contention for the shared wireless medium, not all nodes can concurrently transmit packets to each other in these networks. The maximum number of possible concurrent transmissions is therefore an estimate of the maximum network capacity, and depends on the media access (MAC) protocol being used. We show that for a large class of MAC protocols based on *virtual carrier sensing* using RTS/CTS messages, which includes the popular IEEE 802.11 standard, this problem may be modeled as a *maximum Distance-2 matching* (D2EMIS) in the underlying wireless network: Given a graph $G(V, E)$, find a set of edges $E' \subseteq E$ such that no two edges in E' are connected by another edge in E.**

D2EMIS is NP-complete. Our primary goal is to show that it can be approximated efficiently in networks that arise in practice. We do this by focusing on an admittedly simplistic, yet natural, graph-theoretic model for ad hoc wireless networks based on disk graphs, where a node can reach all other nodes within some distance (nodes may have unequal reach distances). We show that our approximation yields good capacity bounds.

Our work is the first attempt at characterizing an important "maximum" measure of wireless network capacity, and can be used to shed light on previous topology formation protocols like Span and GAF that attempt to produce "good" or "capacity-preserving" topologies while allowing nodes to alternate between sleep and awake states. Our work shows an efficient way to compute an upper bound on maximum wireless network capacity, thereby allowing topology formation algorithms to determine how close they are to optimal. We also outline a distributed algorithm for the problem for unit disk graphs, and briefly discuss extensions of our results to (i) different node interference models, (ii) directional antennas and (iii) other transceiver connectivity structures besides disk graphs.

Keywords: Wireless ad hoc networks, sensor networks, network capacity, MAC protocols, graph theory, approximation algorithms.

I. INTRODUCTION

Embedded wireless sensor networks for monitoring and control applications, rapidly deployable mobile ad hoc networks for emergency and military operation, and "rooftop" networks of nodes connected using radio have all led to an increased interest in networks that have only wireless links. In such multi-hop wireless networks, messages are transmitted via a series of intermediate nodes to their eventual destination. Multi-hop wireless transmission and routing protocols allow individual nodes to have lower power levels and allow the nodes to reuse the same radio frequency (RF) in different parts of the network without significant interference.

Unlike their wired counterparts, nodes in wireless networks that are close to each other in space may not be able to transmit data all at once, because of spatial contention for the shared wireless medium. A media access (MAC) protocol implemented in each node enables wireless nodes to resolve channel contention and avoid collisions.

Because channel contention is a fundamental property of wireless transmission, a natural question to ask is what the aggregate traffic-carrying capacity of a multi-hop wireless network might be. This question has received some recent attention [6]; in practice, the answer is complicated and depends on the MAC protocol used in the network, the directionality of the antennas used, the degree of spatial locality in the end-to-end communication patterns between nodes, etc. In this paper, we shed further light on this problem by considering the problem of determining the *maximum* number of concurrent transmissions at the media access layer that are possible in an ad hoc wireless network. Each such transmission is between nodes that are within radio range of each other, and is a measure of the largest possible network capacity at the MAC layer, since all communication in this case is local. We thus also refer to this problem as he maximum instantaneous MAC layer capacity problem.

The maximum number of possible concurrent transmissions depends on the details of the MAC protocol. A popular MAC protocol today is the IEEE 802.11 standard, which uses *virtual carrier sensing* to resolve channel contention to disallow any other node communication in the vicinity (radio range) of an active exchange. Virtual carrier sensing is a reservation-based scheme—a node wishing to initiate communication to a neighbor broadcasts a "request-to-send" (RTS) message addressed to the neighbor. If the receiving neighbor has not heard of any other on-going transmission, it responds with a "clear-to-send" (CTS) broadcast message. These messages are typically heard by all other nodes within radio range of either (or both) neighbors, and contain information (e.g., packet length) that informs all the nodes of the duration of the data transmission will last. Upon hearing a successful CTS, the initiating node can

---

[1] Basic and Applied Simulation Science (CCS-5), Los Alamos National Laboratory, MS M997, P.O. Box 1663, Los Alamos, NM 87545. Email: `{barrett, anil,marathe}@lanl.gov`. The work is supported by the Department of Energy under Contract W-7405-ENG-36 and in part by the National Science Foundation under Grant No. 0205445.

[2] Department of Computer Science, University of Illinois at Urbana-Champaign. Work done while the author was visiting the Basic and Applied Simulation Science group at Los Alamos. Email: `thite@uiuc.edu`.

[3] Computer Science and Artificial Intelligence Laboratory, MIT. Email: `hari@csail.mit.edu`.

send the data frame; upon successful reception of the frame, the receiving node sends a link-layer acknowledgment (ACK), which informs the sender of a successful transmission. The absence of a link-layer ACK typically triggers a retransmission (usually after an exponentially increasing random backoff); the sender typically retransmits unacknowledged frames a fixed maximum number of times before giving up.

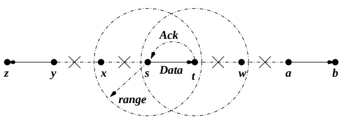


Fig. 1. This figure shows the set of links that can communicate concurrently using virtual carrier sensing (e.g., 802.11). s is the sender and t is the receiver. Since t sends an ACK every time a data frame is received, the neighbors of *both* s and t have to be silent during this time. This implies that the set of edges on which a concurrent successful transmission can occur form a D2-matching. When s and t are communicating, all links marked with × cannot communicate.

Figure 1 shows an example network topology to illustrate the points made above. First, node s sends an RTS message. All of s's neighbor's upon hearing this keep silent for a certain period of time. If t is willing to accept (i.e., it hasn't heard an RTS or CTS recently), then t sends a CTS message. Upon hearing the CTS, all of t's neighbors remain silent for some time (i.e., don't send an RTS or respond with a CTS). Node s now transmits a data frame to t, and t acknowledges successful reception with a link-layer ACK. As a result, during this period of time, s and t's neighbors keep silent.

This paper shows that the maximum possible concurrent transmission problem can be effectively modeled using graph theory, and under reasonable assumptions on the nodes and radios, can be computed efficiently. Given the distribution of nodes in space, and assuming that the nodes are equipped with broadcast radios and omni-directional antennas[4] and implement a virtual carrier sensing (RTS/CTS) MAC protocol, we give an efficient algorithm to compute a bound on the maximum number of possible pairwise node transmissions.

The maximum instantaneous capacity problem is relevant in a variety of practical contexts. For example, a popular approach to saving energy in ad hoc and sensor networks is for nodes to alternate between sleep and awake states. Protocols like Span [3] and GAF [19] attempt to do this while ensuring that the topology induced by "awake" nodes at any time has reasonably connectivity properties; in particular, Span attempts to produce a *capacity-preserving* topology that consumes much less energy than if all nodes were awake. However, there was previously no way to determine how the topologies produced by these topologies compare to the optimal. Our work fills this void. Apart from characterizing the capacity combinatorially, we also give distributed protocols to utilize the channel efficiently (within a constant factor of the total capacity). MAC protocols based on our distributed algorithm can be used to resolve MAC contention. In related work [2], we have modified the 802.11 protocol based on our algorithm, and have obtained promising results.

II. PROBLEM FORMULATION AND RESULTS

A. Models

We introduce the following graph-theoretic model to capture the maximum instantaneous capacity problem. We are given a graph $G(V,E)$, where the vertices of the graph denote the nodes and a (directed) edge (u,v) denotes that u can send a message to v (see Section III-A for how these discrete links may be discerned from radio propagation). Our goal is to choose a subset of edges on which the transmission can occur without conflict: if (s,t) and (s',t') transmissions are occurring simultaneously, then none of the edges $(s,s'),(s,t'),(s',t),(t,t')$ should be present in the interference graph[5]; a pair of such edges $(s,t),(s',t')$ is said to be strongly independent. The set of edges that can be so chosen is called a *D2-Matching*. Formally, given a graph $G=(V,E)$, the problem D2EMIS is to find a D2-matching of maximum cardinality. We will denote the the set of edges in a maximum D2-edge matching in G by $\alpha_2'(G)$. Sometimes, given a set of disks I, we will use $\alpha_2'(I)$, (or just α_2' whenever I is clear from the context) to denote the size of the maximum D2-edge matching for I.

Certain graph theoretic properties of induced matchings make it a capacity-like measure in a network. To see this, let $G(V,E)$ be an undirected graph representing a radio network. Then the following holds: (i) $\alpha_2'(G)$ shows a unimodal variation as edges are added to G: if G is empty then $\alpha_2'(G)=0$; when G is a complete graph $\alpha_2'(G) = 1$, and as edges are added to G, α_2' initially grows for some time, and then starts decreasing, (ii) In a random graph $G(n,p)$, is a concave function achieving a maximum at a point where in the average number of edges is linear in n.

B. Graph theoretic model of radio networks

In this paper, we will model radio networks as geometric intersection graphs. In particular, we use unit disk graphs, disk graphs and (r,s)-civilized graphs as models of radio networks. A disk graph is specified by a set of points V, with a disk $D(v)$ centered at each $v \in V$, with radius $r(v)$. The usual definition of a disk graph, which we call the *undirected model*, (e.g. [10]) is the graph $G(V,E)$, with $e=(u,v)$ if $D(u) \cap D(v) \neq \phi$. This definition does not accurately model radio interference, and so we also consider the following (less common) *directed version*. The directed graph $G(V,E)$ induced by these disks is the following: the set of nodes is V and a (directed) edge (u,v) is present if $v \in D(u)$. The special case where all radii are equal is called a unit disk graph, and in this case, if edge $(u,v) \in E$, then $(v,u) \in E$; as a result we can think of unit disk graphs as undirected. Disk graphs and unit disk graphs naturally model wireless ad hoc networks [10]. The disk around each transceiver corresponds to its broadcast range. A directed edge from node u to node v in the graph captures the semantics that the transceiver corresponding to node u can send a message that can be received successfully by the transceiver corresponding to node v, assuming no

---

[4] In practice, antennas could be directional, and our results extend to this model.

[5] Informally, the interference graph is obtained by inserting an edge from u to v, for all u,v if v lies in the range of u (see Section II-B for details)

collisions occur. Because of our communication model, which requires two way transmission, only bidirected edges can be used for transmission. The unidirected edges only contribute to the interference. However, for the results we discuss in this paper, both models will be essentially equivalent. For ease of description, we will consider the undirected intersection model in Section V and the directed model in Section VI.

Throughout this paper,we will use OPT to denote an optimal solution of an instance I of the D2EMIS problem. By a slight abuse of notation, we will also use OPT to denote the size of an optimal solution and the particular use will be clear from context.

C. Summary of Results

Our main contributions are summarized below.

1. A simple graph-theoretic model of the MAC-induced capacity. The D2-edge matching model is combinatorial in nature and directly captures the MAX-layer interference phenomenon of ad hoc wireless networks. Two previously proposed models for interference are the (i) *protocol model* and (ii) the *physical model* [6]. Our model approximates the protocol model within a constant factor. When the power levels of nodes are non-uniform, the protocol model is no longer valid, but our model still makes sense. In the case of the physical model, our model approximates it within a constant factor for (r, s)-civilized graphs[6].

2. Bounds on $\alpha'_2(G)$ for random unit disk graphs. We consider analytical bounds on the size of $\alpha'_2(G_r)$ in random unit disk graphs G_r: unit disk graphs obtained by uniformly distributing the centers in the plane and assigning each disk a radius equal to r. We show that with high probability $\alpha'_2(G_r)$ is maximized when the radii $r = c/\sqrt{n}$ for some constant c. The resulting matching size is c_1/r^2, for some constant c_1. In fact our result shows how to compute such a matching in almost linear time. The result has several important implications. First, we see that the the size of the matching is maximized before the graph becomes connected. Second, these results are closely related to the recent work of Peraki and Servetto [14] that provides a simpler proof for the Gupta and Kumar result in case of single source case. Our results provide an alternative proof of the theorem in [14] showing that the capacity of network is $\theta(\sqrt{n/\log n})$ for the single source case.

3. Algorithmic results for D2EMIS.

We give efficient approximation algorithms for this problem on geometric graphs. We show that simple sequential greedy algorithms give an $O(1)$ approximation for disk graphs (see Section II-B for its definition). We also extend the sequential algorithms to obtain a fully distributed algorithm with $O(1)$ performance that runs in $O(\log^2 n)$ time. We then turn our attention to devising the best possible sequential approximation algorithms for the D2EMIS problem. Our result is a polynomial time approximation algorithm (PTAS)[7] for disk graphs. We complement these approximation algorithms with computational intractability results. Specifically, we show that unless **NP=ZPP**, D2EMIS cannot be approximated to within a factor of $n^{1-\epsilon}$ for any $\epsilon > 0$. We also show that the problem remains **NP**-complete for graphs that are *simultaneously* planar and unit disk. The hardness results extend the earlier results on this topic by [18], [12].

4. Empirical Results. We experimentally analyze the performance of our sequential and distributed algorithms. For this purpose we use unit disk graphs generated by placing centers of unit disks uniformly at random in a unit square. We also study structured unit disk graphs obtained by placing transceivers in an Urban environment. The following broad conclusions are obtained: (i) the performance of simple greedy algorithms is quite good in practice; typically these algorithms provide solutions that are within a small constant factor (typically 2 to 3) of the optimal solution, (ii) distributed algorithms for computing D2-edge matchings are also quite efficient in terms of quality of solutions and distributed complexity, (iii) the analytical bounds on optimal value of $\alpha'_2(G_r)$ in random unit disk graphs G_r closely matches the empirical bound obtained.

III. RELATED WORK

The complexity of D2EMIS problem was first investigated by Stockmeyer and Vazirani [18]; they called this problem the *induced matching problem*. We prefer to call it the D2EMIS, because this allows us to consider Distance-k matchings for higher k also. Since then the problem has been a subject of active research; we refer the reader to a recent thesis of Mahdian [12] for a detailed account of this problem. D2EMIS is **NP**-hard [18] and in fact the optimization problem was shown to be **APX**-complete in [12] for regular graphs. The problem is known to be solvable in polynomial time for a number of special classes of graphs, including chordal graphs, circular arc graphs, interval graphs, trapezoid graphs and co-comparability graphs (see [5], [12]).

The problem has also been studied for random graphs. See [12], [16] and references therein. The focus of these papers is on fast algorithms exploiting the random nature of graphs and analytical bounds showing that the size of maximum distance-2 matching.

A. Significance of the Distance-2 Matching Model

We describe two widely studied models of MAC level interference (see e.g. [6]) and observe that the D2-Matching model approximates these well. The first model is the Protocol Model: in this model, the transmission from u to v is successful if $\ell(u, v) \leq r$ and any other node w that is simultaneously

---

[6] In this graph class, the minimum distance between two nodes is r, and distance between nodes that are connected is at most s

[7] An approximation algorithm for an optimization problem Π provides a **performance guarantee** of ρ if for every instance I of Π, the value returned by the approximation algorithm is within a factor ρ of the optimal value for I. A **polynomial time approximation scheme** (PTAS) for problem Π is a family of algorithms $\mathcal{F}$ such that, given any fixed $\epsilon > 0$, there is a polynomial time algorithm $A \in \mathcal{F}$ in the family that $\forall I \in \Pi$ returns a solution which is within a factor $(1 - \epsilon)$ of the optimal value for I.

transmitting satisfies $\ell(v,w) \geq r(1+\Delta)$. Typically, Δ is a small constant. Note that this model is well defined only for the case of uniform power levels. The second model is the Physical Model: Transmission from u to v is successful, if

$$\frac{\frac{P(u)}{\ell(u,v)^\alpha}}{N+\sum_{w\neq v}\frac{P(w)}{\ell(v,w)^\alpha}} \geq \beta,$$

where $P(v)$ denotes the power level of node v and α, β are constants related to the radio model and the properties of the antennae of the nodes. Let the capacity in either of these models refer to the maximum number of simultaneous communications that can be performed in a given instance.

Lemma 1: For unit disk graphs, the value of D2EMIS is within $O(1)$ of the capacity in the Protocol Model. For (r,s)-civilized graphs, the capacity in the Physical Model is at most $\frac{1}{\beta}(\frac{s}{r})^\alpha$ times the value of D2EMIS.

It should be clear to observe that the above lemma also holds for the Physical Model if the following constraint is maintained: for transmission from u to v, $P(u)/\ell(u,v)^\alpha$ is bounded. The approximation factor would then depend on this bound.

IV. THE CAPACITY OF A RANDOM DISTRIBUTION OF POINTS

From an information theoretic point of view, the *average capacity* of n points is an important quantity, the average being defined over random configurations of n points in the plane. There is a lot of work (e.g. [6]) that tries to quantify the average throughput, over the MAC and Routing layers together. In this section, we study the variation of the size of D2EMIS with radius, for n points distributed uniformly at random in the unit square.

Lemma 2: Let S be a set of n points distributed uniformly at random in the unit square. For radius $r = \Omega(1/\sqrt{n})$, with high probability the size of $\alpha'_2(G_r)$ for the unit disk graph G_r induced by S is $\Theta(1/r^2)$.

Proof: Place an $r/2 \times r/2$ grid in the unit square. There are $4/r^2$ grid cells. If there are at least 2 points within a grid cell A, a matching edge can be chosen from it. Moreover at most one edge can be chosen from each box. Also, if there are n' grid cells with at least 2 points, $\alpha'_2(G_r)$ is at least cn' for some constant c (pick one grid cell at a time, and discard all grid cells within distance three from it). Therefore, for the lemma, it suffices to show that $\theta(1/r^2)$ grid cells contain at least 2 points, with high probability.

Case 1: $r = \Omega(\sqrt{\log n/n})$. In this case, each grid cell A has $nr^2/4 = \Omega(\log n)$ points, in expectation. Since the choices made by the points are independent, by a Chernoff bound, the probability that cell A has fewer than $c(1-\epsilon)\log n$ points is at most $exp(c\epsilon^2 \log n/2) \leq 1/n^2$, for suitable c, ϵ. Therefore, each grid cell has $\Omega(\log n)$ points, with high probability, and the lemma follows.

Case 2: $c/\sqrt{n} \leq r \leq c'\sqrt{\log n/n}$. Since the expected number of points in a grid cell is smaller than $\log n$, the usual Chernoff bound is not enough to get a high probability bound. We will use the version of the bound from [17]. Let $X(A)$ be a binary random variable that is 1 if there is at most 1 point in the grid cell A. Then, $Pr[X(A)=1] = (1-r^2)^n + nr^2(1-r^2)^{n-1}$. For $r \geq c/\sqrt{n}$, $Pr[X(A)=1] \leq (1-c^2/n)^n + c^2(1-c^2/n)^{n-1} \leq \alpha$, for a constant $\alpha < 1$. Let $X = \sum_A X(A)$ denote the number of cells with at most 1 point. Then $E[X] \leq 4\alpha/r^2$. We now need to get an upper tail bound on X. While it is not true that $X(A), X(A')$ are independent for distinct cells A, A', we nevertheless have $Pr[X(A)=1|X(A')=1] \leq Pr[X(A')=1]$. This is because, if A' is given to have few points, it is more likely that A would have more points, and therefore the probability of $X(A)$, conditioned on this event decreases. From [17], the upper tail Chernoff bound holds even if the variables are not independent, but satisfy the above conditional probability inequality. Now, by applying the Chernoff bound, with high probability, $X = O(4\alpha/r^2)$. Therefore, the number of cells that have at least two points is $\Omega(1/r^2)$, and the lemma follows. ■

Corollary 1: As a function of r, the size of $\alpha'_2(G_r)$ is maximized at $r = \Theta(1/\sqrt{n})$, and the maximum equals $\theta(n)$.

We verify the validity of this bound empirically in Section VII. In our experiments, it seems that the maximum size of $\alpha'_2(G_r)$ is actually very close to $n/4$ and is achieved at $r \simeq 1/\sqrt{n}$.

The result of Peraki and Servetto [14] relies on the following basic fact: a strip in the unit square measuring $1 \times 2d_n$ has a maximum D2-matching of size $1/2d_n$, with high probability, where $d_n = \theta(\sqrt{\log n/n})$ is the threshold radii at which the unit disk graph is connected with high probability. Observe that this easily follows from the lemma above; the lemma also gives a tight bound on the size of maximum D2-matching below this. We should remark that the measure that is computed in [14] is not D2-matching, but in the light of the discussion in Section III-A, can be approximated by it. Our proof is much simpler because of the combinatorial nature of our measure.

V. SEQUENTIAL ALGORITHMS

We first show that a simple gives an $O(1)$ approximation to the D2EMIS problem for disk graphs. We then describe a PTAS for this problem. As described earlier in Section II, for ease of description, we will consider the undirected intersection model for disk graphs in this section. However, the results are also valid in the directed model. Although the methods are simple, Section VII, the algorithms appear to perform quite well in practice. Given the focus of the paper, we have not attempted to improve the theoretical analysis of these algorithms.

For unit disk graphs, the following lemma shows that it suffices to just consider any maximal D2-matching. Let the edges be ordered $e_1, \ldots, e_m$. Consider the greedy algorithm that picks a subset E'. Initially $E' = \phi$. For $i = 1, \ldots, m$, if e_i is strongly independent to all edges in E', add e_i to E'. This algorithm picks a maximal subset of edges. The proof of this lemma is along the same lines as [10], and uses the packing property of unit disk graphs.

Lemma 3: Let G be a unit disk graph, and $E' \subset E$ be any maximal D2EMIS. Then $|E'| = \Omega(|OPT|)$. Therefore, the above greedy algorithm gives an $O(1)$ approximation for D2EMIS problem in unit disk graphs.

We need some more notation for disk graphs. For edge $e = (u, v)$, define $r(e) = r(u) + r(v)$. For node v, let $N_2(v) = \{e = (w, w') | dist(\{v\}, \{w, w'\}) \leq 1\}$, where $dist(A, B)$ is the graph distance between sets A and B in G. For $e = (u, v)$, let $N_2(e) = N_2(u) \cup N_2(v)$. Define $N_{\geq}(v) = \{e \in N_2(v) \,|r(e) \geq r(v)\}$. While Lemma 3 is no longer true for general disk graphs, the geometric packing constraints yield the following lemma.

Lemma 4: ([9]) For any vertex v, the size of the largest distance-2 matching in the subgraph induced by $N_{\geq}(v)$ is $O(1)$.

The greedy algorithm for D2EMIS in disk graphs considers edges in increasing order of the $r()$ values and is given below.

Algorithm D2M-GREEDY-DISK

1) Repeat the following steps until $E(G) = \phi$.
2) Pick edge e such that $r(e) = \min_{e' \in E(G)} r(e')$.
3) Add e to E' and delete all edges $e' \in N_2(u) \cup N_2(v)$.

Lemma 5: Algorithm D2M-GREEDY-DISK gives an $O(1)$ approximation to the D2EMIS problem.

Proof: The proof is by induction on the size of G. When G is of constant size, the lemma is trivially true.

Let the edges picked by the algorithm be $e_1, \ldots, e_r$ (in that order). Since e_1 is an edge with the minimum $r(e_1)$, it follows from Lemma 4 that there are at most $O(1)$ edges of OPT in $N_2(e_1)$. Let G' be the graph obtained by deleting $N_2(e_1)$ and all the incident vertices on edges in $N_2(e_1)$. This implies $OPT(G) \leq OPT(G') + O(1)$. Since G' is a smaller graph than G, the induction hypothesis yields $|\{e_2, \ldots, e_r\}| = \Omega(OPT(G'))$. The lemma now follows. ■

We now describe the polynomial time approximation schemes for the D2EMIS problem for disk graphs. Our algorithm is a direct application of the ideas in Erlebach *et al.* [4]. For unit disk graphs, a simpler and faster algorithm is possible and is based on the ideas in Hunt *et al.* [8]. Due to the focus of this paper, we omit the description of this algorithm.

A. Partitioning the disks

Let $k > 1$ be a fixed integer. (We pick k to be the smallest integer that satisfies $(1 - 2/k)^2 \geq 1 - \varepsilon$.) Apply a scaling transformation to the plane so that the largest disks have diameter 1. Let D be the set of n disks where D_i has diameter $d_i \leq 1$. We partition the set D of disks into *levels* in decreasing order of their diameters. For $0 \leq j \leq l$, level j consists of all disks D_i with diameter d_i in the range $1/(k+1)^{j+1} < d_i \leq 1/(k+1)^j$, where $l = \lfloor \log_{k+1}(1/d_{min}) \rfloor$.

B. Subdividing the plane

We impose a grid on the plane that is the union of $l + 1$ grids, one at each level. The grid at level j for $0 \leq j \leq l$ consists of horizontal lines that are $1/(k + 1)^j$ apart from each other and vertical lines that are also $1/(k + 1)^j$ apart from each other. Therefore, the grid at level j is a subdivision of the plane into squares of side length $1/(k + 1)^j$, and the grid at level $j + 1$ subdivides each square at level j of side length $1/(k + 1)^j$ into $(k + 1)^2$ squares at level $j + 1$ of side length $1/(k + 1)^{j+1}$.

Let $D^{(j)}$ denote the set of disks at level j. The maximum diameter of a disk at level j is less than $1/(k+1)^j$. Therefore, any disk in $D^{(j)}$ intersects at most one horizontal line and at most one vertical line of the grid at level j.

Now consider a subset of the horizontal and vertical lines of the grid at level j defined by two integers r and s where $0 \leq r, s < k$. A horizontal line belongs to this subset and is called *active* if and only if its index modulo k is equal to r, and a vertical line belongs to this subset and is called *active* if and only if its index modulo k is equal to s. In other words, we take every kth horizontal line starting with the horizontal line with index r, and every kth vertical line starting with the vertical line with index s. This operation of taking a subset of the grid at level j is called a *shift* parameterized by the pair (r, s).

The active horizontal and vertical lines of the grid at level j partition the plane into squares whose side length is $k/(k+1)^j$. This is because the grid at level j has line spacing $1/(k+1)^j$ and every kth line is active. Call such a square defined by consecutive active horizontal and vertical lines at level j a *j-square*.

Let $D^{(j)}(r, s)$ be the subset of $D^{(j)}$ obtained by deleting all disks whose interiors intersect either an active horizontal line or an active vertical line in the shifted grid at level j. Let $D(r, s) = \bigcup_{0 \leq j \leq l} D^{(j)}(r, s)$, i.e., $D(r, s)$ is the subset of disks obtained by deleting from D every disk whose interior intersects an active line in the shifted grid at the same level as the level of the disk itself. See Figure V-C.

Lemma 6: For at least one pair (r^*, s^*), where $0 \leq r^*, s^* < k$, the total weight of the maximum distance-2 matching in the subgraph induced by $D(r, s)$ is at least $(1 - 2/k)^2$OPT.

The proof of this lemma follows directly from the methods in [8], [4], and is omitted.

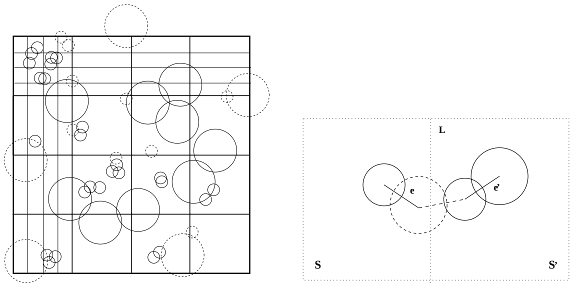
C. Dynamic programming

In this subsection, we restrict our attention to the subset of disks $D(r, s)$ for a fixed pair (r, s) and to computing a maximum D2EMIS in subgraph of G which is the intersection graph of the disks in $D(r, s)$.

Suppose $e = (u, v)$ is an edge corresponding to the intersection of disks D_u and D_v; without loss of generality, assume that D_u has diameter no smaller than that of D_v. By the *level* of the edge e, we mean the level of D_u, the larger of the two disks. A disk at level j is completely contained in a j-square. Let S be the j-square that contains D_u. Then D_v is also contained in the same j-square S. This follows because a grid line that is active for level j, where j is the level of D_u, is also active for all levels k where $k > j$; in particular, it is active for the level of D_v. So, if D_u and D_v were not both contained in S, then at least one of them would cross an active line at their respective level and would have been missing from $D(r, s)$.

We claim that two edges at the same level j belonging to two different j-squares are strongly independent. Let $e = (u, v)$ and $e' = (u', v')$, both at level j, belong to two different j-squares S and S' respectively. Some grid line L that is active for level j separates S and S'. This grid line L is also active for levels greater than j. Assume to the contrary that e and e' are within distance-2 of each other. Therefore, one of $\{D_u, D_v\}$ must intersect one of $\{D_{u'}, D_{v'}\}$. However, this means one of the two intersecting disks must also intersect L. (See Figure V-C.) But this disk would have been deleted since L is active for the level of that disk, which is a contradiction.

Since both D_u and D_v are contained in S, we say that the edge e is contained in S.



(a) (b)

Fig. 2. (a) The outer j-square is subdivided into 16 $(j+1)$-squares when $k = 3$. The disks that intersect active grid lines at their level are shown dotted. The remaining solid disks belong to $D(r, s)$. A few grid lines at level $j+1$ are also shown. (b) Edges e and e' that belong to different j-squares S and S' are d2-independent.

D. The algorithm

We iterate over all k^2 pairs (r, s) with $0 \leq r, s < k$. For each choice of (r, s), we compute the subset $D(r, s)$ of disks and perform the dynamic programming of the previous section. We proceed in decreasing order of levels j where $j = l$, $l-1$, …, 1. For each level j, we consider each j-square S independently.

We enumerate every set I of at most c strongly independent edges of level smaller than j (i.e., incident on larger disks) that intersect S, where $c = O(k^4)$ is a constant. For each such I, we look up in a table the optimal way of extending the set I by adding edges of levels j and higher (i.e., incident on smaller disks) that are contained in S. An entry Table(S, I) in the table corresponding to the j-square S and subset I of strongly independent edges is the optimum set of strongly independent edges that are contained in S and such that $I \cup$ Table(S, I) is an D2-edge matching. By the time the algorithm is ready to compute Table(S, I) it has already computed Table$(S', -)$ for every square S' with level greater than the level of S. The result of the dynamic programming is the union over all 0-squares S of Table$(S, \emptyset)$.

Finally, we choose the set of edges computed by the dynamic programming phase for the value of (r, s) that has the maximum total weight.

E. Proof of correctness & running time

The following Lemma shows that it is sufficient to enumerate the (polynomially-many) subsets I of constant size.

Lemma 7: Suppose S is a j-square and I is a subset of strongly independent edges such that each edge in I has level at most j and intersects S. Then, there is a constant $c = O(k^4)$ such that $|I| \leq c$.

Proof: The proof follows from a disk-packing argument. The disks in I are at level j or less and therefore have diameter greater than $1/(k+1)^{j+1}$. The disks in S are at level j but not at level $j+1$ and therefore have radius at most $1/(k+1)^j$. Let S^+ denote a square of side length $\frac{k}{(k+1)^j} + \frac{1}{(k+1)^{j+1}}$ concentric with S; S^+ is the union of S and a rectangular strip of width $\frac{1}{2}\frac{1}{(k+1)^{j+1}}$ all around S.

For every edge $e = (u, v)$ in I, the area of $D_u \cup D_v \cap S$ is at least half the area of the smaller disk incident on e, i.e., at least $\alpha = \frac{\pi}{4}\left(\frac{1}{(k+1)^{j+1}}\right)^2$. Any two edges $e = (u, v)$ and $e' = (u', v')$ in I must have $(D_u \cup D_v) \cap (D_{u'} \cup D_{v'}) = \emptyset$. Therefore, $|I| \leq$ area$(S^+)/\alpha = (k/(k+1)^j + 1/(k+1)^{j+1})^2/(\pi(1/(k+1)^{j+1})^2/4) = \frac{4}{\pi}(k^2 + k + 1)^2 = c$. ■

The algorithm computes entries in the table bottom-up, i.e., from a smaller square to a larger square that contains the smaller square. When the algorithm terminates, it has considered all disks in $D(r, s)$. When considering a larger square, given that the table entries for all smaller squares contained in the larger square have been computed correctly during the recursive step, the algorithm does in fact correctly compute the table entry for the larger square. This claim is formally stated in the following Lemma.

Lemma 8: Suppose the algorithm computes the table entry Table(S, I) for a j-square S and a set I of strongly independent edges of levels smaller than j that intersect S. Then, Table(S, I) is a strong matching of maximum weight among all strong matchings M that satisfy all the following properties:

1) M consists of edges that are contained in S and have level at least j;
2) $M \cup I$ is a strong matching.

Proof: The proof is by induction on the number of squares processed by the algorithm. When no squares have been processed, we have Table$(S, I) = \emptyset$ for all S, I and the statement of the Lemma holds vacuously.

Now assume that the algorithm processes a j-square S where $0 \leq j \leq l$ and let I be a set of strongly independent edges of levels smaller than j that intersect S. Let I^* be a maximum-weight set of edges of level at least j that are contained in S and such that $I \cup I^*$ is a strong matching. Let $I^*_j \subseteq I^*$ be the subset of edges of level exactly j. Then, $I \cup I^*_j$ is a set of strongly independent edges of level at most j that intersect S. Recall that the algorithm enumerates all such sets.

By the inductive hypothesis, we can assume that Table$(S', I \cup I^*_j)$ has been correctly computed for every $j+1$-square S' contained in S. Also, every edge e in $I^* \setminus I^*_j$ is an edge contained in S and has level strictly greater than j. Furthermore, e is completely contained in a $j+1$-square contained in S; that e cannot straddle two $j+1$-squares follows because all disks that intersect active grid lines have been deleted in $D(r, s)$. Hence, the algorithm, while computing the column Table$(-, I \cup I^*_j)$ of the table, does obtain a table entry with weight at least weight(I^*).

On the other hand, we have shown that two edges belonging to different $j+1$-squares contained in the j-square S are strongly independent and so all table entries computed by the algorithm are indeed strong matchings. Also, $I \cup \text{Table}(S, I)$ is a strong matching. This means that $\text{weight}(I^*) \geq \text{weight}(\text{Table}(S, I))$; therefore, the total weight of edges in $\text{Table}(S, I)$ is in fact equal to the weight of a maximum induced matching in the intersection graph of $D(r, s)$. ■

An immediate corollary of Lemma 8 is that when the algorithm terminates $\bigcup_S \text{Table}(S, \emptyset)$ where the union is over all 0-squares S is a maximum D2-edge matching in the intersection graph of $D(r, s)$.

Theorem 1: D2EMIS has a **PTAS** for disk graphs when represented as a set of disks in the plane.

Proof: Since there are n disks, there can be at most n non-empty squares. Testing whether a given disk belongs to a given j-square is a constant-time operation. For each non-empty square S, the algorithm enumerates sets I of strongly independent edges of cardinality at most c; there are $O(m^c) = O(n^{2c}) = O(n^{O(k^4)})$ such sets. For each S and I, the algorithm performs $O(n)$ table lookups for the $O(n)$ non-empty $j+1$-squares contained in S. A table entry consists of a set of at most m edges; hence, each table lookup takes $O(m) = O(n^2)$ time. The dynamic programming algorithm is executed for k^2 possible values of (r, s). Therefore, the total running time of the algorithm is $O(k^2 \cdot n \cdot n^{O(k^4)} \cdot n \cdot n^2) = n^{O(k^4)}$. Likewise, the algorithm requires $O(m)$ space for each of the $O(n)$ non-empty squares for a total of $O(n^3)$ space.

To verify the claimed performance bound, choose k to be the smallest integer such that $(1-2/k)^2 \geq 1-\varepsilon$; in particular, $k = \lceil (2 + 2\sqrt{1-\varepsilon})/\varepsilon \rceil$ suffices. Then, the approximation ratio of the algorithm is at least $1-\varepsilon$ and its running time is $n^{O(1/\varepsilon^4)}$. ■

VI. DISTRIBUTED ALGORITHMS

A. Distributed computing model

We assume a *synchronous message passing broadcast distributed computing model*. The model is a variation of standard models used in the literature for design and analysis of distributed algorithms that takes into account the broadcast nature of ad hoc wireless networks. In this model, a distributed computing architecture is modeled as a graph with bidirectional links. The nodes of the graph correspond to individual transceivers. Each node has a unique **ID**. The links correspond to radio communication links between individual transceivers. A message by a transceiver is always a broadcast: all the transceivers in its first neighborhood (i.e. with a direct edge to the transceiver) will hear the message simultaneously. We will also assume that each node is aware of the number of nodes in its neighborhood, i.e., its degree. It will turn out that our algorithm can be made to work with an estimate of the degrees, instead of the exact values. Note that we only demand that the knowledge of degrees and not the IDs of individual nodes that are in $n_2(v)$ for any v. We assume a synchronous computing model; each node is assumed to have a clock and these clocks are synchronized. Communication between nodes takes place in *rounds*. In each round, a subset of nodes broadcast a message of length no more than $O(1)$, where n is the number of nodes. The nodes can also do some local computation and hear messages sent by nodes in its neighborhood. If a node u sends a message, then a node v in its range will hear the message successfully unless v also transmits a message or some other node w, s.t. $v \in N(w)$ transmits a message. A collision is said to have occurred if v does not receive the message. Note that the model has two new features as compared to the traditional distributed models used for analysis of algorithms: (i) the broadcast nature of transmissions and (ii) distance-2 interference between nodes as described above. The time complexity of distributed algorithm is given by the number of rounds needed to compute a given task. Although in general distributed computing models ignore the time spent by nodes in each round, all our distributed algorithms spend a constant amount of time in each round.

B. The Distributed Algorithm

Note that by Lemma 3, any maximal distance-2 edge matching is an $O(1)$-approximation for the D2EMIS problem. Therefore, it suffices to construct a maximal distance-2 matching in a distributed manner. Our distributed algorithm is inspired by an elegant distributed algorithm by Luby (see [13]) to compute maximal vertex independent sets using the PRAM model of computation. Our algorithm D2M-DIST-UNITDISK uses a variant of Luby's algorithm described in Peleg [13] (Section 8.4). Figure 3 gives details of our algorithm. There are three main differences between our algorithm and the description in [13].

1) The distributed computing model used in [13] is a *point-to-point* distributed communication model, while our model is a *broadcast model*.
2) The algorithm in [13] requires knowledge of the second neighborhood of each node (or requires messages of size $O(n)$ to be passed, to get this information). We only assume that each node knows its degree. Note that our wakeup probability, and consequently, our analysis are slightly different due to this fact.
3) $\Theta(n)$ size messages need to be passed in [13], while we need to communicate only $O(1)$ size control signals.

1) Analysis: The analysis of the above algorithm follows in the lines of [13], but we need to exploit the geometric structure in order to work with the weaker assumption we make.

Theorem 2: The set of edges $(v, m(v))$ (as defined in Fig. 3) with $\hat{c}(v) = 1$ computed above is a distance-2 independent set of size $\Omega(OPT)$. The above algorithm runs in $O(\log^2 n)$ steps and only transmits messages of size $O(1)$. Thus **Algorithm** D2M-DIST-UNITDISK is a $O(1)$-distributed approximation algorithm for the D2EMIS problem for unit disk graphs.

Proof: The proof is broken into two parts: one for each phase.

Proof for Phase 1: We first show that the set S of nodes at the end of Phase 1 forms a distance-2 vertex independent set and

Algorithm D2M-Dist-UnitDisk

1) **Phase 1:** This phase consists of the following three synchronized steps constituting one round. These steps are repeated until $\hat{b}(v) \in \{0, 1\}$, for each node v. Only nodes v with $\hat{b}(v) = -1$ participate in these steps. Initially, each node v has $\hat{b}(v) = -1$.
    a) Initially $\forall v,\ b(v) = 0$. Each node v set $b(v) = 1$ (wake up probability) with probability $1/(d(v)+1)$. If $b(v) = 1$ then v sends an RTS.
    b) If any node w hears a collision, it sends a COLLISION signal.
    c) If $b(v) = 1$ and v hears no COLLISION or RTS signal(s) from any other nodes in $D(v)$, it sends an RTS-SUCCESSFUL signal and sets $\hat{b}(v) = 1$.
    d) If v hears an RTS-SUCCESSFUL signal from some node in $D(v)$, it sets $\hat{b}(v) = 0$ and retransmits an RTS-SUCCESSFUL signal.
    e) If any node w hears an RTS-SUCCESSFUL signal or a collision due to multiple such signals, it sets $\hat{b}(w) = 0$.
2) **Phase 2:** Let $S = \{v | \hat{b}(v) = 1\}$. Note that S forms a distance-2 independent set. In this phase, for each node $v \in S' \subseteq S$, we choose a node $m(v) \in D(v)$ such that $(v, m(v))$ form a D2-matching.
    Node v maintains a variable $\hat{c}(v)$, initially $\forall v,\ \hat{c}(v) = -1$. For each $v \in S$, the pair v and $m(v)$ work together in the following steps constituting one round. The steps are repeated until $\forall v \in S,\ \hat{c}(v) \in \{0, 1\}$.
    a) v keeps a random variable $c(v)$, initially $c(v) = 0$. Each v sets $c(v) = 1$ with constant probability α. If $c(v) = 1$, then v and broadcasts an RTS1 signal.
    b) For each v, if node $m(v)$ hears an RTS1 signal it rebroadcasts it and sets $c(m(v)) = 1$.
    c) If node v or $m(v)$ sees a collision, due to multiple transmissions, it sends a COLLISION signal.
    d) If $c(v) = c(m(v)) = 1$ and v and $m(v)$ do not hear COLLISION signal, they both send a RTS1-SUCCESSFUL signal, and set $\hat{c}(v) = 1$.
    e) Any node v that hears a RTS1-SUCCESSFUL signal sets $\hat{c}(v) = 0$.

Fig. 3. **Distributed algorithm for solving D2EMIS problem on unit disk graphs.**

takes $O(\log^2 n)$ rounds with high probability. During each execution of step 2(c), if node v sends an RTS-SUCCESSFUL signal, it means that v did not hear a COLLISION signal from any $w \in N(v)$ and therefore, v is the only node in $N_2(v)$ with $\hat{b}(v) = 1$. In Step 2(d), when vertices w in $N(v)$ receive an RTS-SUCCESSFUL signal, they set $\hat{b}(w) = 0$. Every such w retransmits the RTS-SUCCESSFUL signal in Step 2(d). As a result, every node in $N_2(v)$ hears one or more such signals, and in Step 2(e), all such nodes set their $\hat{b}()$ value to 0. Therefore, the new set of nodes added to S in each execution of Step 1 forms an distance 2 independent set, and S is a distance-2 independent set.

We now need to show that Step 1 is run $O(\log^2 n)$ times. Let $A(u, v)$ denote the event that $b(u) = 0, b(v) = 1$ and for $w \in N(u) \cup N_2(v), v \neq w, b(w) = 0$. First, we show that $Pr[A(u, v)] \geq c/(d(v) + 1)$, for a constant c. Let $B(v)$ be a disk around v of radius 2. Observe that both $D(u)$ and $B(v)$ can be covered by a constant number of disks of radius $1/2$ each. Let $S(u)$ and $S(v)$, respectively, be the collections of disks of radius $1/2$ that cover $D(u)$ and $B(v)$, respectively. Now, $Pr[A(u, v)] = \frac{1}{d(v)+1} \prod_{w \in N(u) \cup N_2(v), w \neq v} (1 - \frac{1}{d(w)+1})$, which can be rewritten as $Pr[A(u, v)] \geq \frac{1}{d(v)+1} \prod_{\text{disk } D \in S(u)} \prod_{w \in D} (1 - \frac{1}{d(w)+1}) \prod_{\text{disk } D \in S(v)} \prod_{w \in D} (1 - \frac{1}{d(w)+1})$. Using the fact that for any disk D of radius $1/2$, $\forall w \in D, d(w) \geq |D|$, this can be rewritten as $Pr[A(u, v)] \geq \frac{1}{d(v)+1} \prod_{\text{disk } D \in S(u)} (1 - \frac{1}{|D|+1})^{|D|+1} \prod_{\text{disk } D \in S(v)} (1 - \frac{1}{|D|+1})^{|D|+1}$, Finally, using the fact that $|S(u)| + |S(v)|$ is a constant we get $Pr[A(u, v)] \geq \frac{e^{-\Theta(1)}}{d(v)+1}$. Define event $A(u) = \cup_{v \in N(u)} A(u, v)$. Let Δ be the maximum degree. Suppose $d(u) \geq \Delta/2$, then $Pr[A(u)] \geq \frac{\Delta}{2} \sum_{v \in N(u)} \frac{c}{d(v)+1} \geq c/2$. If event $A(u)$ occurs, u would be not participate in further rounds, and the previous statement shows that for each vertex u of degree at least $\Delta/2$, the probability of this happening is a constant. Therefore, in $O(\log n)$ rounds, all nodes of degree $\Delta/2$ or more would disappear, with high probability. Thus, every successive $O(\log n)$ rounds bring the maximum degree of the residual graph down by a factor of 2. It follows that the number of rounds is $O(\log^2 n)$, with high probability.

Proof for Phase 2: The proof follows closely the proof for Phase 1, except for the fact that nodes v and $m(v)$ are now acting together. Also, by the packing argument used in the proof of Lemma 3, for each $(v, m(v))$, there are $O(1)$ edges $(v', m(v'))$ that participate in Step 3, and are within distance 1 of $(v, m(v))$. Define events $A((v, m(v)))$ with respect to edges, analogous to the definition of events $A(v)$ in Step 1: event $A((v, m(v)))$ holds if there is an edge $(v', m(v'))$ within distance 1 of $(v, m(v))$ such that $\hat{c}(v) = \hat{c}(m(v)) = 0, \hat{c}(v') = \hat{c}(m(v')) = 1$ and for all edges $(w, m(w)), w \neq v$ within distance 1 of $(v, m(v))$ or $(v', m(v'))$ $\hat{c}(w) = \hat{c}(m(w)) = 0$. It is easy to see that $Pr[A(v, m(v))] \geq \epsilon'$ for some constant ϵ'. Therefore, after $O(\log n)$ rounds, each edge $(v, m(v))$ would cease to participate. Thus, Step 3 has to be run $O(\log n)$ times. We only need to verify that the set of edges $(v, m(v))$ resulting after Step 3 forms a maximal distance-2 matching.

The main observation is that if edge $(v, m(v))$ sends an RTS1-SUCCESSFUL signal in some round, it means that and no other edge $(v', m(v'))$ within distance 1 of it became active (i.e., had $c(v') = c(m(v')) = 1$. This implies that the set of edges E' chosen is a distance-2 matching. We now need to show that the size of E' is within an $O(1)$ factor of OPT. While E' need not be a maximal distance-2 matching, it is easy to see that for any node u, some edge $(v, m(v))$ in the disk of radius 2 around u is chosen in E'. Therefore, $|E'| = \Omega(OPT)$. This completes the proof of Phase 2 of the algorithm

Finally, we need to argue about the message complexity. Since we assume that each node knows its degree, there is no need to exchange any messages to figure out the probability with which a node becomes active in Step 1. Also, all the other signals are $O(1)$ size messages. ■

VII. EXPERIMENTAL ANALYSIS

We briefly discuss our experimental results to understand the quality of our capacity model, and the greedy and distributed algorithms for unit disk graphs. Due to space considerations, experimental analysis for disk graphs and (r, s)-civilized graphs will be discussed in a companion paper. The experiments had the following goals: (i) study the variation in the size of $\alpha'_2(G_r)$ with transmission range r (radius), (ii) study the effect of spatial location of transceivers on the size of $\alpha'_2(G)$ in the resulting network G, and (iii) investigate the performance of sequential greedy algorithm and the distributed algorithm.

A. Experimental results and analysis

We summarize and briefly discuss the experimental results. An important note: Given a graph $G(V, E)$, s.t. $|V| = n$, $OPT(G) \leq n/2$. We will use this simple upper bound to evaluate the performance of our heuristics in practice.

1. Unimodal Nature. The capacity of wireline networks can be augmented by adding extra links. The direct analogue of this for ad hoc networks is to increase the radii (power levels) of nodes. This does not always increase the size of $\alpha'_2(G_r)$ in the resulting graphs G_r. In fact, as Figure 4 shows, the size increases up to a certain point, and then decreases. This behavior has important implications for protocol performance and can be seen as follows. Increasing the broadcast range of individual transceivers decreases the average path length between two nodes in the network and results in lower hop count for the routing protocols. On the other hand, increasing the range beyond a certain point decreases the size of the largest D2-matching. Since each successful transmission at the MAC layer uses a feasible D2-edge matching, it is clear that beyond a certain point the performance of the media access layer should go down. Additionally, higher range also implies higher power consumption. The asymptotic maximum value of $\alpha'_2(G_r)$, and the power level (radius) at which it is maximized are important parameters from an information theoretic view (cf [6]). As Figure 4 shows, the maximum size of $\alpha'_2(G_r)$ seems to be about $n/4$, where n is the number of points distributed randomly in the unit square. Moreover, the radius where this maximum is attained appears to be around $c/\sqrt{n}$, where c is either some constant or some very slowly growing function of n (it is hard to figure out which is true from the experiments). For randomly distributed set of points the expected number of points in a region is proportional to the area of the region. This implies that the maximum is achieved when the degree of a node is approximately c_0/n. Compare this to the asymptotic bound on the degree of a node for a random unit disk graph to be connected which is $\sqrt{\log n/n}$. Given the unimodal shape, the best matching size at this degree appears to be $n/\log n$, which is a factor of $\log n$ away from the optimal value.

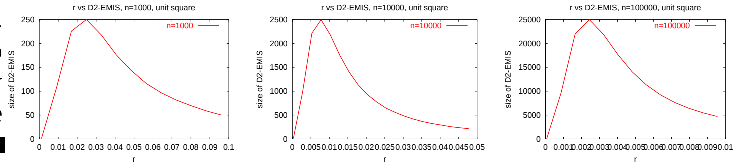


Fig. 4. Variation in size of maximum D2-edge matching with radius, for points distributed randomly in the unit square: the maximum seems to be $n/4$ and is achieved at $r \simeq c/\sqrt{n}$. Note that the solution is no more than a factor of 2 away from optimum.

2. Effect of Spatial Distribution. In recent years, there has been a lot of interest in understanding the real capacity of networks. Starting with [6], several papers have shown via different approaches that the asymptotic throughput of a network is $\Theta(\sqrt{n/\log n})$. All these approaches work with random distribution of points in the plane. In order to test the value of such models, we compare the the size $\alpha'_2(G)$ for random and structured unit disk graphs. A random unit disk graph is constructed by placing transceivers uniformly at random in a plane and assigning each transceiver a radius of r. We also consider unit disk graphs when individual transceivers are located in an urban environment. To generate spatial distributions in an urban setting, we used Portland, OR. We assign transceivers along the roadway system; this roughly corresponds to associating a radio with each car on the road. The random as well as the structured unit disk graphs had about 50 000 nodes in a region measuring approximately $130km \times 110km$.

Figure 5 shows that the random and structured models are quite different. This comparison implies that *optimal parameters for one model are not necessarily optimal for another*.

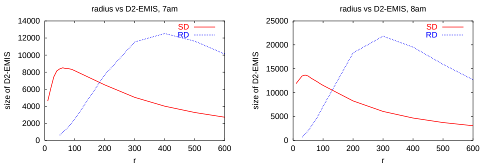


Fig. 5. Comparison between the structured distribution (SD) and random distribution (RD) at snapshots taken at (a) 7am and (b) 8am. The number of nodes is different at the two snapshots- it is about 50000 in (a) and about 87000 in (b)

3. Greedy vs Distributed Algorithms. Our analysis shows that the distributed and greedy algorithms both yield an $O(1)$ approximation to the D2EMIS problem. The exact constant in the approximation guarantee depends on several factors and is hard to estimate. It turns out that the greedy algorithm seems to give a very good approximation (in Figure 4, it is within a factor of $1/2$ of the optimal) for random distribution of points; for other distributions, the approximation might be a slightly larger constant. The distributed algorithm seems to perform slightly worse than greedy by a factor of about 2, as shown in Figure 6. This is quite likely because the solution produced by the distributed algorithm need not be a maximal D2-edge matching. Repeating the distributed algorithm a small number of times, improves the quality of the solution produced by the distributed algorithm. This is illustrated in Figure 6; repeating the algorithm 3 times improved the size of the solution slightly, but beyond that there was no improvement.

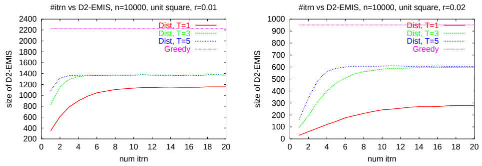


Fig. 6. Comparison of the greedy and distributed algorithms for D2EMIS: the x axis shows the number of iterations for which Step 2 of the distributed algorithm was run. Note that, the size of D2-edge matching computed by the algorithm initially increases rapidly, but slows down after that. T denotes the number of times the distributed algorithm was repeated: in the plots, we show the curves for $T = 1, 3, 5$.

VIII. HARDNESS OF D2EMIS PROBLEM

Lemma 9: For all $\epsilon > 0$, the D2EMIS problem is **NP**-hard to approximate to within a factor of $(n'/2)^{1-\epsilon}$, where $n' = |V'|$.

Proof: The proof is a reduction from Hastad's result [7] on the inapproximability of the independent set within a factor of $n^{1-\epsilon}$, for any $\epsilon > 0$. Let G be a hard instance from [7]. We construct $G'(V', E')$ in the following manner. All the vertices and edges of G are part of G'; all such edges are called *internal edges*. In addition, for each $u \in V$, add a vertex $m(u)$ to V', and edge $(u, m(u))$ to E'. The edge $(u, m(u))$ is called the hanging edge of u in G'. Thus, G' has $n' = |V'| = 2n$ vertices and $m' = |E'| = m + n$ edges.

The lemma will follow from the observation that $\alpha'_2(G') = \alpha(G)$. We argue this by showing that the maximum matching in G' can be modified so that it consists only of hanging edges.

1) If $(u, v) \in E$, then $(u, m(u)), (v, m(v))$ are within distance 2 of each other.
2) If I' is the largest distance 2 matching in G', it cannot have any internal edges: suppose $e = (u, v) \in I'$ is an internal edge in G'. Clearly, the hanging edges of u, v are not part of I'. In addition, for any $w \in V$ adjacent to one of u, v in G (and hence in G'), none of the edges incident on w can be part of I'. As a result, the hanging edges of u, v are not within distance 2 of any edge $e' \in I', e' \neq e$. Therefore, the matching obtained by removing e from I' and adding the hanging edge of u is also the largest matching.

As a result, $\alpha'_2(G') = \alpha(G)$, and approximating $\alpha'_2(G')$ within a factor better than $n^{1-\epsilon} = (n'/2)^{1-\epsilon}$ is not possible, and the lemma follows. ■

The proof can be extended to obtain **NP**-hardness of the D2EMIS problem even when restricted to graphs that are simultaneously planar and unit disk and can be found in the technical report [1].

IX. EXTENSIONS AND CONCLUDING REMARKS

To conclude, our model of capacity can be used to evaluate the performance of MAC protocols in ad hoc and sensor networks. Our algorithmic results can also be used to design better MAC protocols. We have used these ideas to modify the 802.11 protocol and have obtained improvement in the performance and fairness [2]. The modified protocols uses a small amount of local information to implicitly estimate the amount of residual capacity around a transceiver and uses this to set the size of its congestion (back-off) window. These ideas have also been used to develop MAC aware routing algorithms [9].

The above results can be extended in several directions. First, similar techniques immediately can be used for the vertex versions of this problem. Second, the results can be extended when we have more than one available communication frequency. Third, they can be extended so as to apply to other variant interference models and when one uses directional antennas. Finally, our methods can also be used to devise similar approximation algorithms for planar graphs. Some of these extensions are discussed in [1]; other results will be described in companion papers.

REFERENCES

[1] H. Balakrishnan, C. Barrett, V. S. Anil Kumar, M. Marathe, S. Thite. The distance-2 matching problem and its relationship to the MAC-layer capacity of ad hoc networks, Los Alamos Technical Report.
[2] C. Barrett, V.S. Anil Kumar, M. Marathe and S. Thulasidasan, Improving MAC Performance with Locality Information. LANL Technical Report.
[3] B. Chen, K. Jamieson, H. Balakrishnan, R. Morris, Span: an Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks, *ACM Wireless Networks Journal*, 8(5), September 2002.
[4] T. Erlebach, K. Jansen, E. Seidel. Polynomial-Time Approximation Schemes for Geometric Graphs. *Proc. 12th Symp. on Discrete Algorithms (SODA)*, pp. 671–679, 2001.
[5] M. Golumbic, and M. Lewenstein. New Results on Induced Matchings. *Discrete Applied Math.*, 101:157-165, 2000.
[6] P. Gupta and P.R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, Vol. IT-46(2), pp. 388-404, March 2000.
[7] J. Hastad. Clique is Hard to Approximate within n to the power 1-epsilon, *Acta Mathematica*, Vol 182, 1999, pp 105-142.
[8] H. B. Hunt III, M. V. Marathe, V. Radhakrishnan, S. S. Ravi, D. J. Rosenkrantz and R. E. Stearns. NC-Approximation Schemes for **NP**- and **PSPACE**-Hard Problems for Geometric Graphs, *J. Algorithm*, 26 (1998), pp. 238–274.
[9] V.S. Anil Kumar, M.V. Marathe. S. Parthasarathy and A. Srinivasan. End-to-End Packet Scheduling in Wireless Ad-hoc Networks. *ACM Symposium on Discrete Algorithms* (2004).
[10] S. O. Krumke, M. V. Marathe, and S. S. Ravi. Models and approximation algorithms for channel assignment in radio networks. *Wireless Networks*, 7:575–584, 2001.
[11] M. V. Marathe, H. Breu, H. B. Hunt III, S. S. Ravi and D. J. Rosenkrantz. Simple Heuristics for Unit Disk Graphs, *Networks*, 25 (1995), pp. 59–68.
[12] M. Mahdian. The strong chromatic index of graphs, M.Sc. Thesis, Department of Computer Science, University of Toronto, 2000.
[13] D. Peleg. *Distributed Computing: a locality sensitive approach*. SIAM Monographs on Discrete Mathematics and Applications, 2000.
[14] C. Peraki and S.D. Servetto. On the Maximum Stable Throughput Problem in Random Networks with Directional Antennas. Proceedings of the ACM MobiHoc (2003).
[15] S. Ramanathan. A Unified Framework and Algorithm for (T/F/C) DMA Channel Assignment in Wireless Networks, *Proc. IEEE INFOCOM'97*, Kobe, Japan, April 1997.
[16] C. Reidys. Distance-2 Matchings in Random Graphs to appear in Annals of Combinatorics, 2004.
[17] J.P. Schmidt, A. Siegel and A. Srinivasan. Chernoff-Hoeffding bounds for applications with limited independence. SIAM Journal on Discrete Mathematics 8 (1995), no. 2, 223–250.
[18] L. Stockmeyer, and V. Vazirani. NP-Completeness of Some Generalizations of the Maximum Matching Problem. *Information Processing Letters*, 15(1), pp. 14-19, 1982.
[19] Y. Xu, J. Heidemann, D. Estrin, Geography-informed Energy Conservation for Ad Hoc Routing, *Proc. ACM MOBICOM*, July 2001.