Connectivity Maintenance in Mobile Wireless Networks via Constrained Mobility

Joshua Reich, Vishal Misra, Dan Rubenstein
Department of Computer Science
Columbia University
New York, New York 10027
Email: {reich,misra,danr}@cs.columbia.edu

Gil Zussman
Department of Electrical Engineering
Columbia University
New York, New York 10027
Email: gil@ee.columbia.edu

Abstract—We explore distributed mechanisms for maintaining the physical layer connectivity of a mobile wireless network while still permitting significant area coverage. Moreover, we require that these mechanisms maintain connectivity despite the unpredictable wireless propagation behavior found in complex real-world environments. To this end, we propose the *Spreadable Connected Autonomic Network (SCAN)* algorithm, a fully distributed, on-line, low overhead mechanism for maintaining the connectivity of a mobile wireless network. SCAN leverages knowledge of the local (2-hop) network topology to enable each node to intelligently halt its own movement and thereby avoid network partitioning events. By relying on topology data instead of locality information and deterministic connectivity models, SCAN can be applied in a wide range of realistic operational environments. We believe it is for precisely this reason that, to our best knowledge, SCAN was the first such approach to be implemented in hardware. Here, we present results from our implementation of SCAN, finding that our mobile robotic testbed maintains full connectivity over 99% of the time. Moreover, SCAN achieves this in a complex indoor environment, while still allowing testbed nodes to cover a significant area.

I. INTRODUCTION

We focus on a fundamental problem facing mobile wireless networks: How can such a network maintain its own physical-layer connectivity as its constituent nodes move about? Our exploration of *connectivity maintenance* is prompted by such specific examples as the recent DARPA LANdroids initiative to develop a self-configuring network that can deploy itself for temporary use in highly complex wireless environments [1]. More generally, full network connectivity may be required for a network's overall mission (as above), useful for that mission (e.g., coordinated search and rescue, perimeter monitoring), or simply be a way to prevent nodes from becoming lost.

Given the practical nature of our motivation, we focus on designing a protocol that can be implemented on hardware and work in real-world environments. In such environments wireless propagation itself may be quite unpredictable, failing to correlate well with intuitive quantities like distance, because of multipath propagation, interference from outside networks, interference between network nodes, and RF-absorbing environmental features. Even if all of these factors can be suc-

Fig. 1. A testbed node.

cessfully incorporated into a (suitably conservative) predictive model, the presence of obstacles in the environment may prevent actual connectivity from reflecting the model's predictions. Consequently, practical deployment of a connectivity maintenance scheme requires either explicitly mapping the deployment arena, or using an algorithm that does not rely on knowledge of the wireless propagation patterns.

Our work takes this latter approach, as pre-deployment mapping of complex environments is often costly and sometimes infeasible (e.g., battlefield) and online mapping is a challenging problem in-and-of itself [2] (let alone when combined with a real-time connectivity maintenance constraint). To this end, we have developed the *Spreadable Connected Autonomic Network (SCAN)* protocol to handle complex environments in real-time without any prior knowledge of the environment.

Contrastingly, most previous work (e.g., [3], [4]) has relied on simple broadcast models (e.g., deterministic, spherical broadcast). By leveraging geometric properties, tractable algorithms for optimizing the movement of the nodes under a connectivity maintenance constraint can be created. Our work, which does not rely on such broadcast models, cannot produce such optimizations. However, the trade-off is that these approaches cannot address many realistic scenarios: scenarios in which our technique can provide an *implementable working solution*. To demonstrate this claim, we have successfully implemented and tested SCAN on an IEEE 802.11-based robotic wireless networking testbed (Fig. 1). To our best knowledge, SCAN was the first autonomous connectivity maintenance algorithm to have been deployed in hardware [5].

SCAN works by enabling individual nodes to determine when they must constrain their mobility in order to main-

tain connectivity. SCAN achieves this through an entirely distributed process in which individual nodes utilize only local knowledge (2-hop) of the network's topology to *freeze* their movement if SCAN's decision criterion indicates further movement risks network partition. Keeping with our focus on implementability we have designed SCAN to work with commercially available off-the-shelf components. We have also designed SCAN to be essentially agnostic to the particular movement goals of the nodes, allowing SCAN to accommodate differing mission goals. For example, a civilian self-deploying network might aim to maximize coverage area, while a military version may need to balance coverage with providing nodes the ability to move when threatened.

Ideally we would desire to assess our mechanism for connectivity maintenance with respect to how well it provides sufficient flexibility to the network to fulfill that network's particular objective. As these nuances are difficult to capture with a straightforward metric (and hence equally difficult to use as an objective function to be optimized), we assess the performance of our connectivity maintenance mechanism by measuring the *coverage area* it allows while maintaining connectivity with bounded probability. We believe this metric offers a good first-order proxy for our more intuitive but less precise criterion.

While SCAN is only a first step towards a comprehensive solution to an extremely complex and exciting problem, we believe it offers the correct jumping-off point for future research on practical methodologies for connectivity maintenance. Moreover, where previously discussed techniques are applicable, SCAN offers a robust fall-back mechanism, while additionally addressing scenarios in which connectivity of very simple nodes is desired (e.g., micro-scale robots).

Our main contributions are:

- We propose SCAN as a baseline connectivity maintenance mechanism for challenging environments.
- We describe SCAN's intuition and properties under the most general (and challenged) settings. We also show how additional specialized information (e.g., RSSI) can be incorporated into SCAN's framework (Sec. IV).
- Through analysis and simulation, we evaluate SCAN's ability to maintain network connectivity while enabling significant area-coverage. We identify a phase-transition point at which SCAN networks transition between asymptotically frozen and always moving, characterizing that point as a function of the number of nodes and bounding region's size.
- Finally, we implement SCAN on our mobile robotic network testbed as proof-of-concept, and evaluate its performance in providing client coverage (Sec. V).

II. RELATED WORK

Connectivity maintenance for mobile networks has become an active area of research over the past several years. The control theory community began to explore this area first, in the context of motion planning algorithms. In this context, the problems of maximizing some specific target function

and maintaining connectivity are solved jointly. Node movement patterns are determined completely by the specified controller(s). Early work focused on maximal coverage [6] and shortly thereafter on continuously connected group movement [3]. A series of papers considers the use of potential fields to supply centralized [7], distributed [8], and centralized double integrator [4] schemes for ensuring connectivity while maximizing a metric encoded in those fields. More recently, [9] adds the consideration of collision avoidance, and [10] looks at the looser constraint that connectivity reoccur periodically. While most work in this area leverages either geometric properties or assumes perfect knowledge of the potential fields used for determining connectivity and utility, in an approach more closely related to our own, [11] uses only two-hop information to maintain connectivity, albeit by dividing the node population into backbone and regular nodes.

However, all of the above approaches make restrictive assumptions regarding node connectivity - assumptions which are unlikely to be true in practice. By far the most popular such assumption is that node broadcast ranges are perfectly spherical, deterministic, non-interfering, and not subject to attenuation from obstacles or other environmental conditions. Notably, [12] does relax these assumptions, considering a fairly realistic broadcast model. However, this is done at the cost of only being able to consider simple chain topologies. Given this, it is perhaps unsurprising that only a single, preliminary hardware evaluation of a technique from this body of research has been done [13]. Moreover as noted in [13], the evaluation presented could not accommodate cluttered or complex environments (the experiment being conducted in a single, rectangular empty room).

Recently several pieces of work have noted the practical shortcomings of approaches reliant on unrealistically simple broadcast assumptions and have taken heuristic approaches utilizing actual connectivity/signal strength information. Consequently, these approaches have been much more amenable to at least limited hardware prototyping and evaluation. Previously generated radio signal strength maps and hand-produced free space cell decompositions, are used by [14] as input to their connectivity algorithm. A different problem, extending a connected network by having human operators drop "breadcrumb" routers when connectivity begins to weaken, is addressed by [15]. [16] examines mechanisms that repair disconnected networks, leveraging graph properties similar to those used by SCAN. Finally, [17] is the work most similar to our own, developing a distributed algorithm that is a slight modification of the *Neighbor Density (ND)* algorithm we presented previously in [5] and use for comparison here. It is shown in [17] that asymptotically this algorithm will produce a connected network when run in a bounded space. When evaluating this algorithm on a testbed built along the lines published in [18], [17] found eight nodes needed 35 minutes to converge - several times longer than needed by SCAN as described in V-B.

Finally, it is worth noting that, like the body of work above, SCAN does not provide a facility for IP-level routing.

We believe most MANET routing protocols should be able to work alongside the SCAN protocol. However, we note [19] has found that even such protocols may perform poorly with mobile robots. Consequently we recommend choosing a protocol that fits well with the SCAN approach, localizing control messages as much as possible to the vicinity of topological changes.

III. PROBLEM SETTING

A. Operational Environment and Testbed

SCAN was designed to operate in unknown and complex environments using commercially available hardware. As a result, SCAN must contend with unpredictable channel characteristics and unknown obstacles to both wireless broadcast and nodal movement. With respect to the former, wireless broadcast in the 802.11 spectrum is unpredictable, subject to cross-talk, multipath, fading, and interference. These effects are only exacerbated by unknown features of the operational environment. Multipath effects are engendered by walls and obstacles, while fading increases in the presence of RF-absorbent surfaces. In such an environment, knowing where two nodes are *positioned* with respect to one another is often a very different matter than knowing if they will be wirelessly *connected*. Moreover, many environments are GPS denied and while there are techniques addressing indoor localization [20], [21], such systems require significant time to setup and/or leverage expensive hardware.

Automated techniques to create maps that allow such inference to be performed with reasonable confidence are just now being developed [2] and it is unclear if/how these might be incorporated into an algorithm that maps while simultaneously maintaining connectivity. Moreover, such a map can quickly become obsolete should any contributing factor of the environment change sufficiently. Our guiding philosophy behind SCAN is that the most effective and practical measure of whether nodes will be connected in the *future* is their *current* connectivity. Sec. V-A discusses the features of the particular operational environment used for our experiments.

We implemented and tested SCAN on our MADNeT mobile robotic networking testbed, described in [18]. Each mobile node in our testbed comprises a Linksys WRTSL54GS wireless router running a build of OpenWrt Linux affixed to an iRobot Roomba Create mobility and sensing platform (Fig. 1). The WRTSL54GS provides communication, computation and memory, while the Roomba provides power, movement, and environmental sensing. Our nodes do not utilize GPS, although we are working to incorporate RSSI (Sec. IV-E). This setup allows us to experiment with real mobile nodes whose broadcast and mobility decisions we can specify utilizing standardized programming languages.

B. Node Mobility Pattern

Aside from sometimes requiring the Roombas to freeze, we place no other explicit restrictions on their movement. As our primary interest lies in networking, rather than robotics, we have focused our main efforts on SCAN. To spread our nodes across the test area we use a simple movement algorithm. Each

(a) a and b are neighbors. a and d are connected by path (a, b, c, d).

(b) A network with both robust and fragile connections.
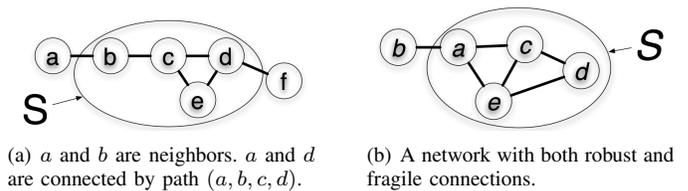
Fig. 2. Network connectivity and robustness.

robot moves straight ahead until it is stopped by an obstacle. Each robot has two sensors which allow it to determine whether the obstacle is off to the side or in front. If the obstacle is to a side, the robot rotates a random angle and continues. If the obstacle is in front, the robot backs up slightly, rotates a random angle, and proceeds. The important point is that the default node movement pattern is *oblivious* to any network connectivity requirement. Consequently, we would believe that SCAN's success, at maintaining connectivity while providing for reasonable area coverage under this movement pattern, generalizes well to many other movement patterns.

C. Network Model

Assume our network contains N mobile robots, each with a (mean) transmission range r, and is deployed in an area A. We represent the network as a graph in which each node corresponds to a mobile robot. Two nodes u and v are *neighbors* and are *directly connected* via a *link*, if they can directly, mutually, and consistently communicate over a wireless channel (if u can receive v's broadcast but not vice versa then u and v are not directly connected).

We define $N(u)$ as the set of nodes that are node u's neighbors with $u \notin N(u)$, and assume, as is the case in our testbed, that each node u knows $N(u)$, and is also informed of $N(v)$ for each of its neighbors $v \in N(u)$. u and v are *connected* if there is a *path* from u to v across a series of links. A network is globally *connected* when every two nodes u and v in the network are connected.

Within each fixed-length *assessment period*, each node assesses its current local connectivity, and, based on this state decides to either *move* or *freeze* until its next assessment. We do not require that nodes make their assessments simultaneously, nor do their respective T's need to match exactly (clock drift is permissible).

Over time, new links can *form* and existing links can *fail*. However, the state of a link between two nodes u and v can change only if at least one of the nodes is moving. If both u and v are frozen, then an existing link between them cannot fail, nor can a link be added when there is none. In other words, only the mobility of node pairs significantly alters whether a pair can communicate.

IV. ALGORITHMS FOR CONNECTIVITY MAINTENANCE

Our goal is to develop and evaluate a basic mechanism that intelligently leverages knowledge of current network connectivity characteristics to assess the robustness of current network connectivity. This assessment is then used to determine whether further movement endangers future network connectivity. If so, we require that the node(s) for whom this is true refrain from further movement by freezing until

such time as continued movement no longer poses this risk. To this end we will introduce two algorithms leveraging this basic mechanism: a naive *Neighbor Density (ND)* algorithm which uses a very simple metric for assessing connectivity robustness, and SCAN which conducts a still simple, yet significantly more powerful assessment of robustness.

Both ND and SCAN share the common assumption that, while the quality of the wireless channel may fluctuate unpredictably over space, it will remain relatively constant over time: relative movement of two nodes may effect their connectivity, but time-wise fluctuations will not have a significant effect. In situations where the quality of the wireless channel is fluctuating wildly over time (e.g., significant and varying external interference, fast fading) more specialized or conservative techniques will be required.

A. Neighbor Density Algorithm

The *Neighbor-Density (ND)* algorithm (shown in Fig. 3(a)) serves as a naive parameterizable heuristic solution to the connectivity problem. ND utilizes nodal density (or more precisely valence) to achieve connectivity: if a node has more than k neighbors it considers local connectivity robust and consequently may move, fewer it must freeze. Both [15], [17] use slight variations on ND for maintaining connectivity. ND's messages are constant in the number of neighbors, as only the sending node's ID need be sent.

B. Spreadable Connected Autonomic Network Algorithm

The *Spreadable Connected Autonomic Network (SCAN)* algorithm (shown in Fig. 3(b)) takes the greedy approach that a node's movement should only be constrained in direct response to a perceived lack of robustness in the (local) network connectivity structure. To get a feel for when we may wish to freeze a node, consider the example in Fig. 2(b).

Node a is connected to 3 neighbors. To disconnect a from any of the neighbors to its right (or in fact any of the nodes to which it is connected) at least 2 links in S must be broken. In contrast, only a single link needs to fail to disconnect node a from node b. When links fail infrequently (e.g., 1 failure during a assessment period), then if a were only concerned about the nodes in the set S, it could continue to move. However, there is a high likelihood that any movement could cause the single link between a and b to fail, ending the connection with b, thereby partitioning the network. To keep the network partition-free, both a and b should freeze.

But what if, during a assessment period, $k \geq 1$ links can be expected to fail? How do we then determine whether nodes must freeze to ensure network connectivity is maintained? It is this question that SCAN is designed to address.

Formally, SCAN is pre-configured with a parameter k, such that a node u is allowed to move as long as $|N(u) \cap N(v)| \geq k$ for every $v \in N(u)$. In other words, u moves if it shares k neighbors with each of its neighbors v. If this property does not hold for even a single neighbor, u must freeze.

SCAN does require more messaging overhead than ND. SCAN's messages are linear in the number of neighbors, since the sending node must send its own ID and all those of its neighbors. In practice however, if density is high enough that this should become an issue, the probability of disconnection becomes very low. A practical implementation could switch over to ND until density falls.

C. Global Connectivity

Consequently, if a pair of nodes who are connected do have sufficient redundancy in that connection (as routed through mutual one-hop neighbors), SCAN concludes that movement on either of their part may in the worst case sever all known local routes between these two nodes. Other longer routes may in fact exist should all local paths be severed, in which case these nodes might still remain connected, but SCAN conservatively assumes that only known connections can be relied upon in assessing connectivity robustness. By assessing connectivity robustness between each pair of nodes on a local basis, we argue (informally) that SCAN prevents any pair of directly connected nodes from severing all of the locally known paths between them, and transitively prevents global network partition.

While we lack space here to make this argument rigorous, we refer the interested reader to our tech report [5] which proves SCAN's global connectivity property and shows that for any node to be disconnected from a SCAN network at least $k + 1$ local link failures must occur within one SCAN assessment period.

D. Choosing SCAN's k Parameter

Since SCAN disconnections only occur when at least $k+1$ nearby links fail simultaneously, the best value of k depends upon how many neighboring links are expected to fail (i.e., nodes move out of communication range) within a assessment period. As one increases the speed of a node, decreases the range of transmission, or increases the broadcast cycle time, a larger k is needed to ensure connectivity. In general, as k increases, it becomes less likely that the network will partition, but the expected time that nodes spend moving is reduced as well, which can delay achievement of the goal for which mobility of nodes is required in the first place. In Section VI we find that for network whose nodes move slowly $k = 2$ is more than sufficient while for more volatile settings $k > 4$ appears extremely robust.

Pre-determining the optimal k for an arbitrary setting is quite hard because: (i) link failures will often exhibit co-dependence, (ii) disconnection of a nearby nodes are dependent, (iii) $k + 1$ nearby link failures permit *but do not necessitate* severing of all locally known paths between two neighbors, and (iv) even if all locally known paths connecting a node to some neighbor are severed, there may still be a global path connecting them.

Consequently, we will use a back-of-the-envelope calculation to give some insight into how k should be set. We begin by setting the probability p of a link breaking during a given SCAN assessment period to the distance a node travels in a period divided by the mean broadcast radius. Roughly speaking, $k + 1$ or more links break with probability order p^{k+1}. Assuming a disconnection lasts for a mean time of τ assessment periods, the expected fraction of time the network is fully connected is $1 - \tau p^{k+1}$. k should be chosen such

$$\textbf{if } |N(u)| \geq k \textbf{ then move else } \text{freeze}$$

(a) ND

$$\textbf{if } |N(u) \cap N(v)| \geq k, \ \forall v \in N(u) \textbf{ then move else } \text{freeze}$$

(b) SCAN

Fig. 3. Mobility criteria.

that τp^{k+1} is less than the tolerated level of partitioning. The interested reader may observe our algorithm in action and test the effect of changing k using a simplified applet version of our simulation environment [22].

E. Incorporating Additional Information

While we do not address utilizing additional sources of information such as GPS or RSSI in this paper, we do wish to briefly describe how they might be incorporated in SCAN's general approach. SCAN simply views each connection as a binary value, 1 if connected, 0 if not, corresponding to the presence or absence of an edge in our network. A version of SCAN utilizing additional information (SCAN+) could use weighted edges, whose weights correspond to normalized RSSI values or relative distance measurements. One possibility would be to require that the weighted sum of the of the paths connecting any neighbor and a given node be greater than a constant k' allowing the presence of strong connections to offset lower absolute numbers of common neighbors. Clearly more sophisticated schemes could also be devised leverage an increasingly nuanced view of the connectivity topology for improved performance. We leave these for future work.

V. Testbed Experiments

Our was to develop connectivity maintenance techniques that could actually be implemented and tested on hardware in a noisy, challenging environment. To this end, we have implemented SCAN on our Roomba robotic testbed and run several hours of experiments. This allowed us to validate our ideas in a practical setting and demonstrate SCAN's efficacy in maintaining the connectivity of a self-deploying mobile network. Recall from Sec. III-B that our nodes operated in a GPS-denied environment and explored the area randomly. Clearly, if our blindly moving nodes could stumble into a successful configuration, nodes with more robust mobility routines tailored for a particular application could do at least as well.

We found that SCAN could provide connectivity in a remarkably robust fashion while also providing latitude of movement sufficient to cover clients scattered throughout our test environment. Out of 273 minutes and 50 seconds of experiments, our network remained connected in all but 2 minutes and 23 seconds. Moreover, the network partitions we encountered were comprised of one node disconnecting, with the sole exception of a 15 second period during which a pair of nodes partitioned themselves as a connected component.

A. Experimental Setup

Our experiments were run on the 8th floor of Columbia University's CEPSR research building, covering approximately $1900m^2$. A dozen or so wireless networks were competing for use on this particular floor, providing a moderate level of interference. As previously discussed, we assess SCAN based by measuring the *coverage area* it allows while maintaining

connectivity with bounded (in this case 99%) probability. Since measuring the total coverage area of our combined nodes with instrumentation was not feasible in our test environment, we instead placed wireless clients around the floor and measured the total number of clients covered as Fig. 4(a) illustrates.

While the experimental space was moderately large and subject to both wireless interference from competing networks, as well as broadcast obstacles, our nodes could still broadcast a good proportion of its length. To conserve power and evaluate our algorithm in a more transmission-limited environment, we dialed down the broadcast power to the minimum level supported in software ($0.25dBm$) and did not restrict the shielding of client nodes (e.g., if they were behind doors, in far corners, or on the ground).

Our experiments tested networks of size $N = \{4, 5, 6, 7, 8\}$ using $k = 2$. Each data point averages 10 trials.

To choose this value of k we followed the logic laid out in Sec. IV-D. Given Roomba speed ($0.5m/s$), SCAN assessment cycle every $3s$, mean broadcast radius $20m$, and mean number of cycles for disconnection $\tau = 100$: we have $p = 3(s) * 0.5(m/s)/40(m) = 0.088$ and $\tau p^{k+1} = 100 * 0.088^3$ which would imply a tolerable partition likelihood of around 5% for $k = 2$. In fact we found partition frequencies noticeably lower than this in our experiments.

B. Experimental Results

In our tests SCAN maintained full network connectivity over 99% of the time. As can be seen from Fig. 4(b), this high degree of connectivity maintenance did come at the cost of constraining area coverage. In this figure the left y-axis measures the percentage of trials in which coverage was achieved, the right y-axis the time in minutes, and the x-axis the number of nodes. Networks of 4 nodes were unable to ever fully cover all nodes at the same time. Yet nodes were still able to move far enough that every client was covered for at least some significant proportion of the experiment as seen in Fig. 4(c) which plots the percentage of the time a given node was covered (y-axis) against the value of k used (x-axis). We can also see that the client nodes were quite heterogeneous with respect to coverage: certain clients were always covered, while others were often quite difficult to cover, which intuitively makes sense given the area's complexity.

For $N > 5$, we see a significant performance improvement. In this space a network of 6 nodes seems sufficient to provide simultaneous coverage to all nodes, although it takes over 2 minutes to do so. We see a continuing decrease in the time taken until all nodes are covered as N increases, along with an increase in success rate. The decrease in performance for $N = 7$ is most likely due to a high variance resultant from the statistically smaller number of trials run. Thus, SCAN coupled with the most rudimentary of mobility mechanisms enabled a small number of mobile nodes to self-organize a configuration
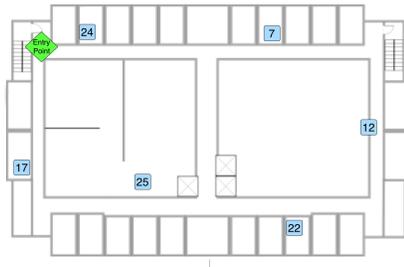
(a) Indoor space/clients used for experiments.
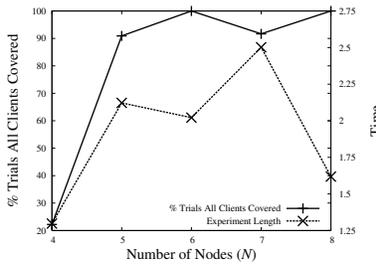
(b) % Trials coverage achieved/time taken.
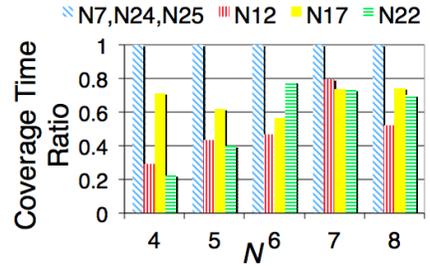
(c) Fraction of time clients were covered.

Fig. 4.   Testbed experiment results.
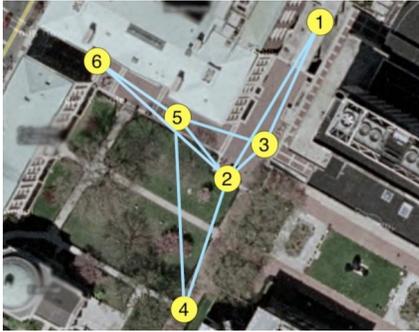
Fig. 5.   Aerial view of frozen configuration, 6-node experiment.

capable of covering to all clients.

C. Additional Results

We also conducted a preliminary assessment of our system on several outdoor areas. A more thorough assessment of our SCAN's function outdoors is currently in the planning phase and will likely incorporate a comparison of basic SCAN with an implementation of SCAN+ incorporating GPS and/or RSSI (Sec. IV-E).

Our outdoor assessment revealed two very interesting things. The first was that even using our basic hardware setup with broadcast power set to the minimum allowable, our testbed was still able to spread over a significant area as can be seen in Fig. 5. The resultant frozen configuration for the six nodes spanned an area over 100m north-south and 70m east-west. The second was the degree to which the environment effected connectivity/effective broadcast range. Only 300m north on an adjacent plaza we ran the same experiment, but the average distance between nodes was less than 33% that of our previous location. We believe these differences are in large part due to the differing levels of radio frequency interference in these two locations (a weekend test showed greater distance spread between nodes at the northern location). Video of our experiment is available at [22].

VI. SIMULATION AND ANALYSIS

To further develop our understanding of SCAN and its scalability, we turn to simulation and analysis. In this section we seek to answer three essential questions regarding the behavior of networks running SCAN:

- Under what conditions will all nodes in a network running SCAN ultimately freeze?
- How well does a network running SCAN, when frozen, cover an area as a function of the probability of retaining connectivity?

- What is a good value of k to use?

While the latter two questions have fairly straightforward motivation, the first question bears additional discussion. This question is a significant one since if our nodes are confined to a very small space relative to their broadcast capabilities, then sufficient disconnections will never occur to force all to freeze. Sometimes this could prove highly inconvenient (one would not want a civilian self-deploying network moving continually underfoot), while in others it might prove helpful (the same behavior in a military setting might help nodes avoid being targeted by the enemy). Either way it is valuable to understand this phenomenon.

We investigate asymptotic freezing via analysis, which we then confirm by simulation in an obstacle-free environment. Further, through simulation we find that SCAN outperforms ND by a significant but not overwhelming margin in obstacle-free environments. However when we introduce walls and use a realistic physical-layer wireless model to our simulation (an environment matching SCAN's design concerns), we see that SCAN continues to perform well for the same range of k values used in the obstacle-free environment, while ND's performance decreases drastically.

A. Simulation Platform and Assumptions

All the simulations discussed in this section were conducted on NetLogo 4.0.2 [23], a combined Logo-like language and simulation platform. Netlogo is ideal for modeling a distributed protocol whose behavior influences and is influenced by the topology of the dynamically evolving network upon which it is running. Unless otherwise noted, all plot points average 100 trials.

1) Obstacle-Free Environment: In our obstacle-free simulation, nodes roughly within broadcast range r of one another were considered connected. To provide additional realism to our simulations, we varied the range of broadcast stochastically. In this simulation two neighbors are connected if the distance between them is less than or equal to a normal random variable with mean $r = 1$ and $\sigma = 5\%$. Each potential pair of neighbors had its own independently chosen random variable, and these random variables were regenerated at a rate equal to the frequency of the movement decision made by the nodes We ran these experiments for $N = \{25, 50, 100, 200\}$.

2) Indoor Environment: Our indoor simulation mode deals with an environment with walls of different types. For modeling the signal propagation in that environment, we used the

COST 231 Multi-Wall Model (MWM) [24, Ch. 4]. Among the empirical models, this is one of the most sophisticated ones and it is applicable in the 2.45GHz band. The MWM model provides the path loss as the free space loss added with losses introduced by the walls and floors penetrated by the direct path between the transmitter and the receiver. Since we consider a single floor, the loss (in dB) is given by:

$$L = L_{fs} + L_c + k_{w1}L_{w1} + k_{w2}L_{w2} \qquad (1)$$

where L_c is constant loss (we assume $L_c = 0$), k_{w1} is the number of light walls, $L_{w1} = 3.4$dB is the loss due to a light wall, k_{w2} is the number of heavy walls, $L_{w1} = 6.9$dB is the loss due to a heavy wall, and L_{fs} is the free space loss given by

$$L_{fs} = 32.4 + 20\log(r/1000) + 20\log(f) \qquad (2)$$

where r is the distance between the transmitter and receiver (in meters) and f is the frequency in MHz (2450 in our case).

As much as possible, we aimed for our simulation to parallel our indoor experiments, using the same set of barriers shown in Fig. 4(a) for our indoor simulations. We set the transmit power of a node to $0dBm$ ($1mW$). We assume that the receiver sensitivity is $-82dBm$, which is a reasonable value for ICE 802.11g devices and we assume a fast fading margin of $16dB$. Hence, we require that the propagation loss will satisfy $0 - (-82) - L > 16$ (or simply $L < 66$) in order for two nodes to be within transmission range. In an environment without walls this translates to allowed distance of approximately 17m. In a multi-wall environment the distance varies with the locations of the different nodes.

We note that the simulation model ignores collisions between SCAN messages simultaneously sent by other nodes. In our actual implementation, messages were sent on the order of seconds making such collisions very unlikely.

Given the higher computational complexity of these experiments our simulations here were done for $N = \{8, 16, 32, 64\}$.

3) Link Failure: In either simulation environment, the main factor in whether two nodes are connected arises from nodes' mobility. In our analysis, we do not consider stochasticity, relative movement being the only factor in determining connectivity. This assumption is made only to ease our analysis and is not needed for SCAN to function correctly, as our testbed experiments in Sec. V demonstrate.

B. The Freeze Phase-Transition

If our nodes are confined to a very small space relative to their transmission radii, then they will never freeze. As the size of the space is increased, and they are able to spread further apart, the likelihood of freezing increases. As the size of the space grows to ∞, we eventually will reach a point where, with probability 1, all nodes will have frozen. At this point, since all nodes are frozen, they cannot obtain new neighbors, and the network remains in a frozen configuration. Here, we investigate the point of the phase-transition: for a given k and N, what is the *ratio* of the size of the space to the node transmission radius, beneath which the network is forever moving, and above which the network always reaches
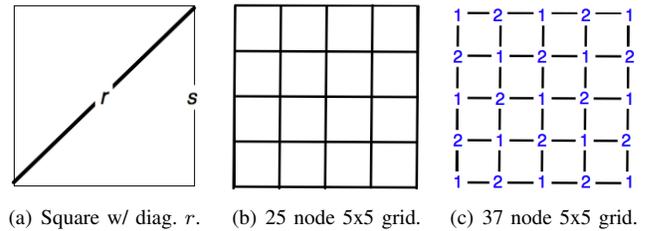
(a) Square w/ diag. r. (b) 25 node 5x5 grid. (c) 37 node 5x5 grid.
Fig. 6.   Geometry of our bounding area model.

a freezing configuration? We will first build a model to predict this point and then verify our model's accuracy via simulation.

1) Minimum Bounding Area: To determine the inflection point, we begin by considering for a fixed broadcast radius and number of nodes how tightly those nodes could possibly be packed and still freeze. By closely bounding how tightly these nodes might be packed, we can then reverse this relationship and come to an approximation of how many nodes might be packed in a given area and still freeze.

Our model is composed of two components: a regular spatial pattern in which nodes can be laid out in a frozen configuration and the minimum scaling of this spatial pattern below which additional links will form. If we choose our model appropriately, the minimum area needed by this model will approximate the minimum area needed for such a system to freeze. Our main task is to identify a regular spatial pattern that is more dense than almost any frozen configuration we can expect to encounter and then to deliver a closed form equation for that pattern's size as a function of k, N, and r.

We begin by considering the simplest case $k = 1$ and a simple topology, a perfectly square space of area A. In order for a network to freeze, we must find some spatial configuration of the nodes such that no node has any neighbors in common with any other neighbor, the nodes are configured as densely as possible, and the configuration is regular enough to analyze easily. The final criterion leads us to explore *regular tessellations*. A *tessellation* is created when a shape is repeated over and over again covering a plane without any gaps or overlaps. A *regular tessellation* is simply a tessellation composed of *regular polygons* - polygons for which all sides are the same length s. As it turns out, our search is relatively simple since there are only three regular polygons which tessellate in the euclidean plane: the triangle, square, and hexagon [25].

Triangles cannot be used since any neighbor v of a given node u is also neighbors with the third node w on any triangle built upon edge u, v. Of the two remaining options, the hexagon allows for a tighter packing. However, it is moderately more difficult to work with than the square, upon which, as we will see, a very reasonable approximation of the phase-transition can be built.

Examining Fig. 6(a) we see that the maximum size for s in a square, the length of whose diagonal is $> r$ is $r/\sqrt{2}$ since $r^2 = 2s^2$. We can then place one node on each of the grid points on a $\lceil\sqrt{N}\rceil X \lceil\sqrt{N}\rceil$ grid as seen in Fig. 6(b). Such a grid will take up an area of $A = (r^2/2)(\lceil\sqrt{N} - 1\rceil)^2$ and the

ratio of one node's broadcast to the N-node bounding area is

$$\pi r^2/A = 2\pi(\lceil \sqrt{N} \rceil - 1)^2 \qquad (3)$$

Extending this model to cover larger k is not overly difficult. For $k = 2$ instead of placing a single node at each grid intersection, we place two nodes at every other intersection as seen in Fig. 6(c). In this way, each node shares precisely one node with any of its neighbors, whether it is alone on its grid point or sharing it. Then, for a given N we only need $\lceil \sqrt{2N/3} \rceil - 1$ grid lines on each side, requiring an area of $(r^2/2)(\lceil \sqrt{2N/3} \rceil - 1)^2$. $k = 3$ is even easier requiring us to put two nodes at each grid intersection. We can extend this strategy to arbitrary $k \geq 1$ obtaining:

$$\pi r^2/A = 2\pi/(\lceil \sqrt{2N/(k+1)} \rceil - 1)^2 \qquad (4)$$

which describes the ratio of an individual node's broadcast area to the total area. As will now be seen, our model's prediction tightly bounds the behavior seen in simulation.

2) Ratio of Broadcast Area to Bounding Area in Simulation: Our simulation results in this section were obtained through a binary search for the largest ratio of individual node broadcast area to bounding area that would result in a frozen configuration. The precise phase-transition point is difficult to determine via simulation (since showing the hypothetical transition value fails to ever freeze would require infinite time). Instead, we estimate this value by measuring the average convergence times from our experiments in the unbounded space and allowed our system to run in excess of 10 times the maximum convergence times taken there.

Each combination of N and k received 10 trials, each over the course of up to 5000 time-steps. To obtain a clearer correspondence with our model, in this trial alone we did not stochastically vary the connection lengths. At the end of each trial that did not result in a freezing configuration, the ratio was decreased by half its current value for the subsequent trial. Conversely, whenever a trial ended in a freezing configuration the ratio would be increased by half. The first trial began with a ratio above the point where freezing could occur, but not too far. This was determined by a set of preliminary experiments.

The results of our exploration for $k = 1, 4$ are shown in Fig. 7(a), as are the model predictions from (4) (other k values bound similarly but were omitted for graphical clarity). In this figure the y-axis measures the ratio of an individual node's broadcast area to the total bounding area while the x-axis measures the number of nodes N, each plot point representing the highest such ratio found at which a network of N froze.

The behavior of the phase-transition point for all k can be characterized roughly as for every doubling in the number of nodes, the ratio between node broadcast area and bounding area decreases by slightly less than half. Moreover, increasing values of k appear to lie at a relatively constant log-scale distance above one another. This is unsurprising as at higher levels of k a given number of nodes with a given broadcast area can fit into a smaller bounding space while still being able to reach a freezing state as argued in our model above.

C. Performance in Obstacle-Free Environments

Here, we explore properties of the frozen configuration when SCAN and ND are applied in an unbounded space, where the configuration is guaranteed to eventually freeze. In our simulation, nodes are deployed at a gateway from which they proceed to spread across a 2-dimensional plane. As these nodes spread across the plane, they extend the area which the network attached to the station covers. However, if nodes becomes cut-off from the gateway through a network partition event, the entire area over which only these nodes broadcast ceases to be covered. Here we measure *coverage area* as the union of the area over which nodes currently connected to the gateway can broadcast. Consequently, coverage implicitly takes into account global connectivity, insofar as that connectivity benefits the coverage goal of a self-deploying wireless network.

Fig. 7(b) plots the size of SCAN's coverage area (y-axis) as a function of k (x-axis). The different curves depict differing numbers of nodes in the network, with each node's communication range averaging a unit distance. Not surprisingly, the coverage area increases in proportion to the size of N, and is a decreasing convex function with the size of k, where nodes are required to maintain larger collections of neighbor sets. Additionally, it is worth noting that for $k = 1$ the benefit of increased mobility in providing greater coverage is more than offset by the decrease in connectivity. While for $k > 2$, the frequency of any network partition does decrease but proves increasingly costly from a coverage standpoint.

Fig 7(c) provides a comparable plot for ND. The same trends discussed for SCAN are apparent, although optimally parametrized SCAN covers in excess of 50% greater area than optimally parametrized ND.

We lack the space for an in-depth investigation of connectivity and partition rate as a secondary phenomenon separate from coverage. For such an investigation see [5].

D. Performance in Indoor Environments

In this set of simulation experiments we examined how SCAN and ND performed in a more complex environment, filled with walls that were obstacles to both wireless signal propagation and also to node movement. For SCAN we found the relationship between k and the coverage area to be substantially similar to those of the obstacle-free environment, albeit with slightly higher rates of partition for a given k. However, things are very different for the parametrization of ND. In an indoor environment the presence of walls affects not only connectivity but also movement. Indoors, corridors and other obstacles increase the likelihood that a cluster of nodes begins moving in the same direction. When this occurs ND's behavior becomes pathological. Particularly if the number of nodes in the cluster is greater than k, then all nodes in the cluster will be able to move away from the rest of the network without ND's freezing criterion being triggered. Thus k must be made very high in order to maintain connectivity.

Partially as a consequence of this, partially because SCAN's functioning is little affected by obstacles, the difference in performance between them is significantly greater than in

| (a) Freezing phase transition. | (b) Attained coverage, SCAN. | (c) Attained coverage, ND. | (d) Indoor, 32 nodes. |
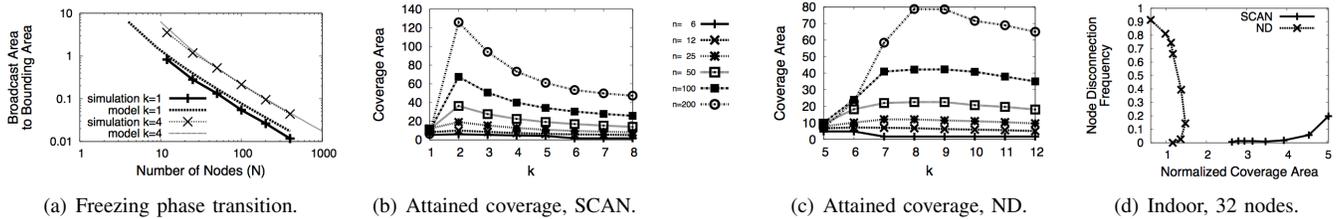
Fig. 7. Simulation results.

our obstacle-free environment. In Fig. 7(d) we can see a comparison of the performance of both SCAN and ND for a system of 32 nodes (other values of N plot out similarly). The x-axis of this figure plots the normalized coverage area and the y-axis plots the frequency of disconnections. Several aspects of this plot are noteworthy. Firstly, even at its worst parametrization ($k = 1$), SCAN provides strongly bounded connectivity of 80%, while poorly parametrized ND provides almost no assurance of connectivity. Secondly, for a given minimum level of required connectivity, SCAN far outperforms ND, generally covering between 2 and 3 times greater area for the network sizes studied in this simulation experiment. Finally, in this plot, one can see that ND may produce the same area coverage for different node disconnection frequencies. When a low k is chosen, ND does a poor job at maintaining connectivity and no nodes are attached to the base, resulting in low coverage despite nodes traveling relatively far, while a high k causes the network to freeze before it has covered much area.

E. Summary of Results

In this section, we identified the freezing phase transition point for networks running SCAN. Above this point the network will freeze and below this point the network will not. We find that our analytical approximation is a good fit for our simulated results. We then focused our attention on obstacle-free environments. Using simulation, we found SCAN superior, covering an area 1.5 times larger than ND. Finally, we examined SCAN's behavior under more realistic simulation conditions, finding that its behavior vis-a-vis optimal k remained substantially the same, but that its performance advantage over ND increased five-fold over that found in obstacle-free environments.

VII. Conclusion and Future Work

In this paper, we have presented SCAN, a fully-distributed, low-overhead, tunable protocol for maintaining the physical layer connectivity of a mobile wireless network. SCAN enables nodes to move about the space as they desire, freezing *only* when risk of further movement endangers the network's connectivity. By relying on local connectivity information instead of localization data combined with predictive wireless models, SCAN is able to handle noise and obstacles present in realistic settings.. SCAN is *extensible*, allowing for the inclusion of additional information (e.g., RSSI), and *expects little* from the environment or hardware, making it an ideal connectivity maintenance mechanism for challenged hardware/environments, or to serve as a backup mechanism for higher-performance techniques with stronger environmental/hardware requirements. When tested on hardware in a

challenging indoor environment, we found SCAN allowed for significant area coverage while robustly maintaining full network connectivity over 99% of the time.

References

[1] "DARPA LANdroids project." [Online]. Available: http://www.darpa.mil/ipto/programs/ld/ld.asp

[2] J. Fink and V. Kumar, "Online methods for radio signal mapping with mobile robots," in *Proc. IEEE ICRA*, May 2010.

[3] D. Spanos and R. Murray, "Motion planning with wireless network constraints," *Proc. IEEE ACC*, June 2005.

[4] M. Zavlanos and G. Pappas, "Distributed connectivity control of mobile networks," in *Proc. IEEE CDC*, Dec. 2007.

[5] J. Reich, V. Misra, D. Rubenstein, and G. Zussman, "Spreadable connected autonomic networks (SCAN)," Columbia University, Tech. Rep. CUCS-016-08, Mar. 2008.

[6] D. Spanos and R. Murray, "Robust connectivity of networked vehicles," in *Proc. IEEE CDC*, Dec. 2004.

[7] M. Zavlanos and G. Pappas, "Controlling connectivity of dynamic graphs," in *Proc. IEEE CDC-ECC*, 2005.

[8] ——, "Potential fields for maintaining connectivity of mobile networks," *IEEE Trans. Robotics*, vol. 23, no. 4, pp. 812–816, Aug. 2007.

[9] N. Bezzo and R. Fierro, "Tethering of mobile router networks," in *Proc. IEEE ACC*, June 2010, pp. 6828 –6833.

[10] G. Hollinger and S. Singh, "Multi-robot coordination with periodic connectivity," in *Proc. IEEE ICRA*, May 2010.

[11] Z. Yao and K. Gupta, "Backbone-based connectivity control for mobile networks," in *Proc. IEEE ICRA*, May 2009.

[12] C. Dixon and E. Frew, "Controlling the mobility of network nodes using decentralized extremum seeking," in *Proc. IEEE CDC*, 2006.

[13] N. Michael, M. Zavlanos, V. Kumar, and G. Pappas, "Maintaining connectivity in mobile robot networks," in *Experimental Robotics*, 2009.

[14] M. A. Hsieh, A. Cowley, V. Kumar, and C. J. Taylor, "Maintaining network connectivity and performance in robot teams," *Field Robotics*, vol. 25, pp. 111–131, January 2008.

[15] M. Souryal, A. Wapf, and N. Moayeri, "Rapidly-deployable mesh network testbed," in *Proc. IEEE GLOBECOM*, 2009.

[16] N. Atay and B. Bayazit, "Mobile wireless sensor network connectivity repair with k-redundancy," in *Algorithmic Foundation of Robotics VIII*, ser. Springer Tracts in Advanced Robotics, G. Chirikjian, H. Choset, M. Morales, and T. Murphey, Eds. Springer Berlin / Heidelberg, 2009, vol. 57, pp. 35–49.

[17] N. Correll, J. Bachrach, D. Vickery, and D. Rus, "Ad-hoc wireless network coverage with networked robots that cannot localize," in *Proc. IEEE ICRA*, May 2009.

[18] J. Reich, V. Misra, and D. Rubenstein, "Roomba MADNeT: a Mobile Ad-hoc Delay Tolerant Network Testbed," *ACM MC2R*, Jan. 2008.

[19] F. Zeiger, N. Kraemer, and K. Schilling, "Commanding mobile robots via wireless ad-hoc networks: A comparison of four ad-hoc routing protocol implementations," in *Proc. IEEE ICRA*, May 2008.

[20] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements," in *Proc. ACM SenSys*, 2004.

[21] N. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System," in *ACM MobiCom*, Aug. 2000.

[22] [Online]. Available: http://www.cs.columbia.edu/~reich/research_connectivity.php

[23] S. Tisue and U. Wilensky, "Netlogo: A simple environment for modeling complexity," in *ICCS*, 2004.

[24] D. Cichon and T. Kurner, "Euro-cost 231 final report," Tech. Rep., 1998.

[25] D. Chavey, "Tilings by regular polygons—ii: A catalog of tilings," *Computers and Mathematics with Applications*, no. 17, 1989.