Measuring performance of ad hoc networks using timescales for information flow

Raissa M. D'Souza, Sharad Ramanathan, and Duncan Temple Lang
Bell Laboratories, Lucent Technologies
Murray Hill, NJ 07974
Email: raissa@microsoft.com, {sharadr,dtemplelang}@lucent.com

Abstract— We define metrics to characterize the performance of ad hoc networks based on timescales for information flow, power consumption and interference. The statistical distribution of timescales has not been previously considered. Yet, it is important for understanding the feasibility of communicating over such networks, for comparing different algorithms for building up network topology and for distinguishing regimes of routing. We quantify the longest timescale for information flow and estimate its distribution. We also introduce a decentralized adaptive power algorithm, that uses only information local to each device, for building ad hoc networks. This algorithm is shown to perform significantly better by all our metrics when compared with a standard, constant power, algorithm.

I. INTRODUCTION

Understanding how a collection of wireless devices that know only of their local environment, can organize into a communications network with no central control is an important open problem. Such ad hoc networks are both dynamic and temporary since the network topology changes as devices move in space, as new devices join the region and others leave, and as devices turn on and off. By sending out queries and listening for replies, devices can learn the identity of other devices in their transmission range and hence the local network connectivity. Knowledge of the topology beyond the immediate transmission range is conveyed along a sequence of intermediary devices. In this manner devices build up knowledge of their connectivity, potentially storing the information in address books and routing tables. Communication with distant regions relies fundamentally on devices cooperating in relaying one another's data. Thus a message may "hop" from device to device when following a path from source to destination. Such networks operate entirely through peer-to-peer interactions and could be of use in a variety of situations from mobile military units to a collection of moving cars transmitting information about upcoming road hazards. For an overview of some outstanding research issues and potential applications, see for instance [1].

What kinds of algorithms would individual mobile communication devices use to build up such a network amongst themselves? How do we compare the relative performance of such algorithms? Performance can be measured primarily by the maximum achievable throughput given a constrained set of resources. We measure this indirectly in terms of three performance metrics: the power consumption, the interference from other users and the time to transmit messages. All three quantities are statistical in nature and their respective distribution functions rely upon the properties of the underlying network, characterized by the user density, usage patterns, routing strategies, network topology, etc. Thus to measure performance, one must first carefully quantify each of the performance metrics and determine techniques to estimate their statistical distributions. Then the statistical advantages of different network building algorithms can be compared. While the power usage and interference are easier to quantify[2], more care is needed in determining the statistics of a characteristic time to transmit messages, defined herein by the estimated time needed to diffuse information throughout the network. Aside from being a performance metric, the statistical distribution of this characteristic time tells us the feasibility of building such networks and furthermore gives insight into what size to build routing tables and how often to refresh them.

Conventional networks are for the most part static. Thus complete routing tables can be built and used to efficiently direct packets. In dynamic ad hoc networks, routing tables expire after a period of time since the network topology changes with time. If routing tables are to be used with such mobile networks, how often should one refresh them to update changes in network topology? It is known that if nodes are "fast" moving, data in routing tables will quickly become obsolete; the spatial location of the nodes will change significantly in the time it takes for a message to hop from source to destination. In contrast, if the nodes are "slow" moving, routing tables will persist for some amount of time. In the extreme limit where the nodes are stationary, routing tables will persist for all times. But "fast" and "slow" are not absolute quantities. They are defined only relative to other timescales of interest, such as the characteristic time for message delivery. Thus on a network with a short characteristic time (*i.e.*, in which data exchanges occur rapidly), airplanes can appear to be slow moving. On a network with a long characteristic time, people can appear fast moving. To quantify the relations amongst the various timescales for mobile, wireless, peer-to-peer networks we need to understand: (1) the density of nodes and of traffic in the network; (2) the relative speed of the mobile devices; and (3) the relative speed of data flow across the network. Quantifying the distinct timescales will be especially important for networks with heterogeneous clients, where different types of clients may move with greatly varying velocities.

The aim of this paper is to define and measure the afore-mentioned performance metrics. These metrics are then used to compare networks built by a well known constant power level algorithm and those built by a decentralized adaptive power algorithm. We consider ad hoc networks made of a collection of identical nodes, distributed at random with density ρ, in a two-dimensional space. We introduce a framework for analyzing the timescales for flow of information, in a *dense* traffic regime, in the limit where the nodes are *stationary*, assuming that information diffuses on the network. Though, in practice, an optimized scheme for exchanging data will be used, this simplification of diffusion makes it possible to easily estimate various timescales for information flow. We are interested especially in the longest timescale associated with diffusive flow, τ, as it gives an estimate of the time required to learn the full network topology (*i.e.*, build a complete routing table) via a sequence of queries and replies which diffuse along the network. This τ is the measure we use to define the characteristic time for information to diffuse throughout the network.

The *distribution* of τ is obtained by studying many in-dependent realizations of networks with the same fixed user density. The feasibility of building such networks hinges on this distribution having a small enough variance. Past work has assumed the diffusion of information over mobile ad hoc networks to derive results on their capacity[3]. For such models to be feasible, it must be shown that the characteristic time to deliver messages does not fluctuate drastically if the network topology changes. Yet this issue of feasibility and the distribution of characteristic times, though extremely relevant, has not been previously addressed.

As mentioned, we also use the distribution of τ as one of the performance metrics for quantitatively comparing different schemes for building up an ad hoc network from a collection of initially isolated devices. The well-studied common power (CP) level model for constructing an ad hoc network[4], [5] assumes all devices transmit at the same power level. We introduce a an adaptive power (AP) level construction, which uses directional information, similar to [6]. This is an iterative scheme where each device sets its power individually and adaptively, using only local information. We directly compare the performance metrics of the CP and the AP schemes. For each instance, the AP construction has a more efficient topology (and hence a smaller value for τ), smaller overall power consumption for both the typical and extremal integrated power, and reduced interference. In addition, since each device sets its operating power level individually based on the local environment, the overall network can quickly adapt to changes. Furthermore, by optimizing based on geometric connectivity, rather than minimizing power at each node, some nodes operate at higher power than by the CP scheme. However such nodes introduce "shortcut" paths through the network. The latter two issues, of adaptation and shortcuts, make the AP scheme especially well suited for use along with routing algorithms, such as ad hoc on demand distance vector routing[7], which require continual execution of route discovery algorithms.

This manuscript is organized as follows. Section II, is largely an extension of past work, included to clarify the problem formulation. We first describe the CP approach, then determine the minimum power requirements for full connectivity for an ensemble of independent realizations of networks, and extract the scaling behavior as the number of nodes is increased yet their spatial density, ρ, held constant. Section III introduces the issue of timescales and the matrix formalism used throughout the remainder. A connectivity matrix that specifies which nodes are directly connected to which others is constructed. Those direct connections define a static graph and we study a diffusive dynamics on that graph. The dynamics can be described by a state transition matrix P. We are in particular interested in the eigen-spectrum of P. The eigenvalues determine the timescales for information flow, and the corresponding eigenvectors can be used to identify simple bottlenecks. In Sec. IV we introduce our AP construction, and compare it to the CP construction using the performance metrics discussed above.

II. CONNECTIVITY AND THE COMMON POWER LEVEL CONSTRUCTION

We first determine the requirements for building a fully connected network of stationary nodes, that all transmit at a common power (CP) level. Consider N devices initially distributed uniformly at random in a two-dimensional space of area $L \times L$ (thus the spatial density of devices $\rho = N/L^2$). The coordinates of the ith device are denoted by $\vec{x}_i$, and the spatial distance between the ith and jth devices, $d_{ij} = ||\vec{x}_i - \vec{x}_j||$. The power level of a transmission decreases on average with distance from its source, so the magnitude at the source determines the spatial range, R, over which the signal strength will be distinguishable from noise. The CP assumption has analytic and practical advantages as discussed in [8]. In addition to reducing collisions of transmissions, it ensures reciprocity—if transmissions from the ith device are perceptible by the jth, those from the jth are perceptible by the ith. Thus if $d_{ij} \leq R$, the devices are two-way "connected" and exchange messages directly. If $d_{ij} > R$, messages can be relayed between the two nodes only if there is a connected path of intermediaries.

Routing messages would be trivial if each node broadcast at a large enough power to communicate directly with all other nodes. Yet power is a limited resource. Furthermore, the broadcast nature of wireless means a transmission interferes with all other simultaneous transmissions, having the greatest impact on those in its range R. The desire to minimize interference and power consumption means we want the transmission range to be the smallest possible while still ensuring full connectivity. We denote the value of this "critical range" by R_c. With the CP scheme, all nodes broadcast at this corresponding power level. This model was introduced in the context of multihop communication networks by Gilbert in 1961, and loose bounds on R_c obtained[4]. Recently strict bounds on R_c have been obtained in the asymptotic limit,
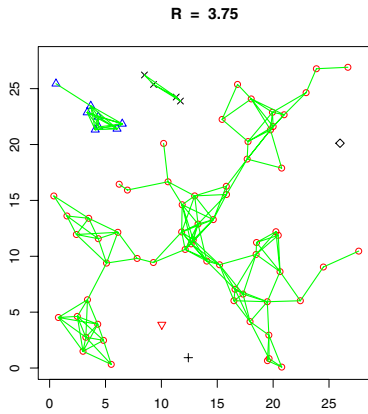
Fig. 1. We illustrate a typical realization for a network consisting of $N = 78$ nodes at a spatial density $\rho = 0.1$ (hence $L = 28$). The nodes are initially distributed uniformly at random and are illustrated by the various symbols scattered in the plane. Nodes which are connected, for $R = 3.75$ as shown, form a cluster and are represented by the same symbol, with solid lines illustrating the direct connections. For this value of R, the system has $c = 6$ disconnected clusters.
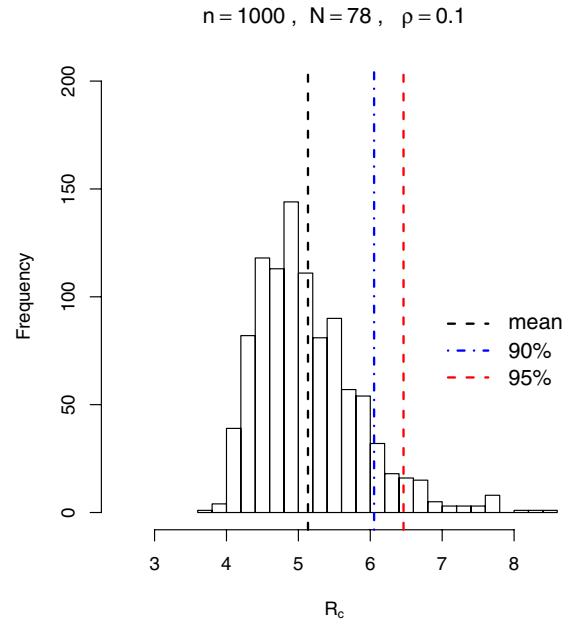

Fig. 2. A histogram of the frequency with which a given value of R_c occurred for $n = 1000$ realizations of networks with $N = 78$ processors at density $\rho = 0.1$. The vertical lines denote respectively the mean, $\langle R_c \rangle$, the 90th percentile and the 95th percentile. This gives us a clear idea of the distribution of R_c in the non-asymptotic regime.

where the density ρ approaches infinite (and hence N, the number of devices, also approaches infinite)[5]. The proof uses techniques similar to those for percolation theory and the theory of coverage processes (*i.e.*, covering a two-dimensional unit circle with disks of fixed size). We are interested, however, in the more physically realistic regime where N is small and finite, and understanding how R_c scales with increasing N for fixed ρ. We consider a range of finite values for N, for each of two different densities, $\rho = 0.1$ and $\rho = 0.2$, and generate $n = 1000$ independent realizations of networks for each set of parameters. Each realization is distinguished by the random locations of the nodes. We determine R_c for each realization, as explained in detail in Sec. III-B; essentially we iteratively construct a connectivity matrix for a given value of R, and use the eigenvalues of the matrix to determine connectivity. We also determine the distribution of R_c over all these realizations for each $\{N, \rho\}$-pair studied. Fig. 1 illustrates a typical realization. $R_c = 5.45$ for this realization.

Figure 2 is a histogram displaying the frequencies of the different R_c values observed in the $n = 1000$ realizations with $N = 78$ nodes and density $\rho = 0.1$. The results for $\rho = 0.2$ are similar, though rescaled, and thus not included here. The dashed vertical lines mark respectively the mean (referred to from here on as $\langle R_c \rangle$), the 90th percentile and the 95th percentile. Note the finite support indicated by the upper tail with rapidly decreasing density. This distribution in the non-asymptotic regime had not been previously determined, yet it is important. If ad hoc networks are to be built they will consist of a finite number of nodes, initially starting with on the order of tens of nodes. The information in such distributions could be used as a starting point for building networks with processors randomly distributed in space using the CP construction.

Moreover, having determined the distribution of R_c for a range of finite N's and different ρ's, we want to know if

this has predictive power: if it is possible to extract a scaling function for how the average value for a given number of nodes and density, $\langle R_c(N, \rho) \rangle$, varies as we vary N. Building up connectivity is analogous to building the minimal spanning tree of a collection of nodes: the minimum range for full connectivity R_c is analogous to the length of the longest edge necessary to complete a connected tree. It is well known that the length of the longest edge decreases from the asymptotic value as $\ln(N)/N$, see for instance [9]. Assuming this form, we do a one parameter fit of our data for $\langle R_c(N, \rho) \rangle$ to the function $\langle R_c(N, \rho) \rangle = R_\infty(\rho) \left[1 - \ln(N)/N \right]$. Here the estimated parameter is $R_\infty(\rho)$, the asymptotic value of R_c as N approaches infinite for a given ρ. In Fig. 3 we plot $\langle R_c(N, \rho) \rangle / R_\infty(\rho)$ for various values of N, for $\rho = 0.1$ and $\rho = 0.2$. The dotted line is the curve $y = 1 - \ln(N)/N$, the theoretically expected behavior. It adequately describes the empirical data, capturing the general trend in a simple way.

III. QUERY TRANSMISSION IN THE HIGH TRAFFIC LIMIT

In addition to spatial connectivity requirements, temporal ones are also relevant. For instance, how long would it take for the nodes to determine if they are fully connected? For each realization studied above we first determine R_c (who's distribution is summarized in Fig. 2). Recall the nodes are stationary, hence the network topology fixed. Once connectivity is established we consider a dynamics for the flow of information on this static network. The largest timescale associated with this flow estimates the time required to learn the full network
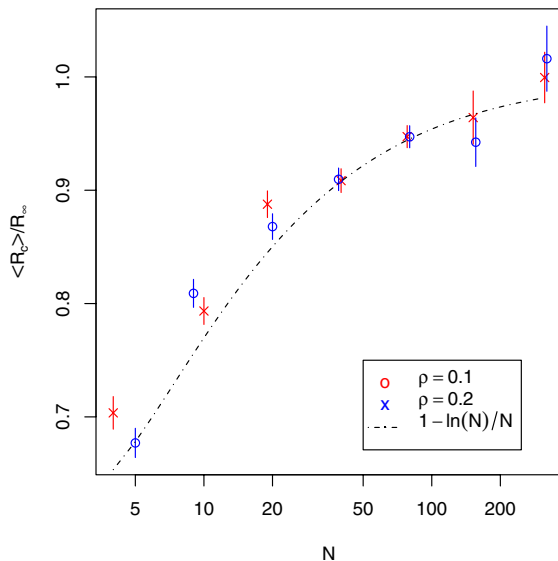
Fig. 3. A scaling function for $\langle R_c(N,\rho)\rangle / R_\infty(\rho)$ versus N, for two values of ρ. The error bars represent the standard error over all the independent realizations, n. For the first five points $n = 1000$. For the largest two points $n = 100$. We use this data to estimate the asymptotic value $R_\infty(\rho)$, and find $R_\infty(\rho = 0.1) = 5.42 \pm 0.04$ and $R_\infty(\rho = 0.2) = 3.85 \pm 0.05$. The solid line is the theoretically expected behavior, $\langle R_c(N,\rho)\rangle / R_\infty(\rho) = 1 - \ln(N)/N$.

topology and thus build a complete routing table (*i.e.*, the time for the final message carrying new information of the topology to be received). In addition, it sets a reference point. Processes occurring during time intervals much longer than this can be treated as approximately stationary during durations of time less than or equal to this reference interval.

We make the simplest approximation for the dynamics: a data packet located on a node may take a random walk step from that current node to one of its directly connected nodes. This assumption simplifies the model considerably and allows us to quickly and easily estimate timescales. It corresponds to a diffusion process on the graph. When making this assumption we need to understand the connection between random walks and broadcasts over wireless channels. Broadcast communication means a data packet could conceivably be communicated to all neighbors during one transmission. Yet our approximation assumes that at any time at most only one neighbor is in a state of being ready to receive the packet (the others being occupied exchanging messages with other devices). And thus this corresponds to a regime of dense network traffic. In addition, wireless transmission means simultaneous broadcasts interfere with one another. As discussed below, we weight the transition probabilities for the random walk to reflect these additional effects of interference, and thus more accurately capture the essence of wireless data transmission. The assumption of packets taking random hops to a connected neighbor is a worst case scenario

with regards to efficiently transmitting data. With an actual ad hoc network, we would use some strategy for efficiently exchanging messages. Thus we are establishing upper bounds on the time to send data in a regime with dense network traffic.

A. State transition matrix

We can model the random walks of the data packets using a matrix formalism, where the matrix specifies the transition probabilities for the walkers. The eigen-spectrum of the that matrix tells us modes of behavior and associated timescales. As discussed below, we must first establish a connectivity matrix, then adapt it to incorporate a simple model for interference, to obtain the state transition matrix.

The direct connections between the nodes in the network specify the elements of the connectivity matrix, M. If nodes i and j are directly connected matrix element $M_{ij} = 1$, otherwise $M_{ij} = 0$. Note the diagonal elements $M_{ii} = 1$, so nodes are connected to themselves. We are considering a discrete time random walk process, executed synchronously across the network. During each discrete update of the network, each data packet will choose at random amongst one of these direct connections (including the one to itself) and accordingly hop to an adjacent node or remain stationary. For instance if $M_{ij} = 1$, a data packet on node i would have some probability P_{ij} to hop to node j during the next discrete update. We can easily modify the connectivity matrix M to obtain the state transition matrix, P, specifying these P_{ij}'s. In the simplest case, the probability to hop to any connected neighbor is equally weighted, as is the probability to stay stationary:

$$P_{ij} = M_{ij}/\sum_j M_{ij} = M_{ij}/k_i. \qquad (1)$$

Note, $k_i = \sum_j M_{ij}$, is the number of direct connections for node i (*i.e.*, the edge degree of node i, including the self link). So for the equally weighted case, $P_{ij} = 1/k_i$ if $M_{ij} = 1$, and $P_{ij} = 0$ if $M_{ij} = 0$.

We intentionally allowed for the random walkers to remain stationary since this provides a mechanism for incorporating interference. We make the simple assumption that if any node in your neighborhood is transmitting, you cannot transmit. Thus if a node is connected to k_i others, it can only transmit on average $1/k_i$ of the time, at which point it would send out a data packet. So, viewed in terms of packets, the probability for a data packet located on a specific node to remain stationary on that node,

$$P_{ii} = (k_i - 1)/k_i. \qquad (2)$$

When the packet takes a step every $1/k_i$ updates, it hops with equal probability to any of the other (k_i-1) directly connected nodes;

$$P_{ij,i\neq j} = 1/\left[k_i(k_i - 1)\right]. \qquad (3)$$

These transition probabilities are illustrated in Fig. 4, where we show one node connected to k others and the probability to hop along the various links.
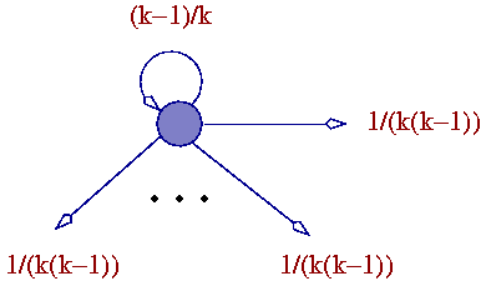
Fig. 4. Edge weighting to approximate the effects of interference. We show a node of the network which is connected to k other nodes (including the self link). A data packet located on this node would hop along one of the links during the next discrete update of the space. The probabilities for following each link are shown adjacent to the link. This uses the approximation that a node connected to k other nodes only transmits $1/k$ of the time.

This simple approach of edge weighting approximates to first order the effects of interference. But it neglects transmissions that fail due to the hidden terminal problem[10], and also the connectivity of the $(k_i - 1)$ adjacent nodes. We over-estimate the interference caused by heavily connected adjacent nodes, as they will transmit less often than the estimated rate of $1/k_i$. And we underestimate the interference caused by sparsely connected adjacent nodes, which will transit more frequently than estimated. Recall this is in the regime with dense traffic so nodes want to transmit as often as possible. A second order approximation, taking into account longer range interactions, should be more accurate. We should still be able to use distance-based truncation of interference effects. Evidence has been published elsewhere that such models with simple truncation adequately reflect interference in wireless communications systems[11]. We would like also to incorporate results for interference from multicasting on trees[12], and multiple antennae noise cancellation schemes[13].

B. Timescales and modes

Given a set of node locations and a value for R, we can compute the corresponding connectivity matrix M, and thus also the transition matrix P. Once P is established for such a realization, we study its eigenvalues and eigenvectors. We are interested in the modes of behavior associated with the dynamics described by P. By definition if $\vec{v}_i$ is an eigenvector of P, with associated eigenvalue λ_i,

$$P\vec{v}_i = \lambda_i \vec{v}_i. \tag{4}$$

And applying the state transition matrix t-times yields:

$$P^t \vec{v}_i = (\lambda_i)^t \vec{v}_i. \tag{5}$$

Note, we obtained P by independently normalizing each row of a symmetric matrix (so the dynamics described by P conserves probability, meaning no random walker is created or destroyed). So all the elements of P are real and less than or equal to unity, likewise the eigenvalues are all real and less than or equal to unity. Each eigenvalue λ_i has an associated eigenvector v_i. If the network were composed of N nodes, v_i would be a N-dimensional vector. The j-th component of v_i

indicates the state of the j-th node with a real number that can be positive or negative (which we call the amplitude of the j-th node). An eigenvalue $\lambda_1 = 1$, has an associated eigenvector v_1 which describes a steady-state solution to the dynamics. In steady-state, the amplitude at each node no longer changes with subsequent evolution under P. If the network of nodes is fully connected, there is one unique steady-state solution, so only one $\lambda_i = 1$. The remaining eigenvalues are all less than unity, hence describe decaying modes (*i.e.*, initialized in a state described by such a mode, the amplitudes at each node decay). If the network is not fully connected, and instead consists of independent sub-clusters, there will be a steady-state solution for each sub-cluster. Using this approach, we determine R_c; it is the smallest spatial range, R, for which only one eigenvalue is unity. For a discussion of state-transition matrices and their eigen-spectrum see for instance [14].

We are interested in the timescales associated with the decaying modes, and in particular with the most slowly decaying mode. We define the "relaxation time", T, for a mode, as is standard in the physics literature, as the time for the amplitude of the mode to decay by a factor of $1/e$. Thus $P^T \vec{v}_i = \vec{v}_i / e$. Hence the longest timescale in our system, τ, is the relaxation time for the slowest decaying mode (denoted by $\vec{v}_p$):

$$P^\tau \vec{v}_p = (\lambda_p)^\tau \vec{v}_p = \vec{v}_p / e. \tag{6}$$

Equivalently,

$$\tau = -\frac{1}{\ln(\lambda_p)}. \tag{7}$$

This longest timescale is associated with the second largest eigenvalue, λ_p, (*i.e.*, the "penultimate" eigenvalue, which is the one closest to yet less than one). We can interpret the positive and negative amplitudes as corresponding to two different types of viscous fluids, and the relaxation time as the time required for the fluids to mix.

C. Simulation Environment

Applying this framework involves extensive numerical simulation and statistical analysis of results. We wanted a software environment that would allow for rapid high-level prototyping, accessibility to a rich collection of statistical tools, good visualization of results, and flexibility to easily extend the model in the future. The R language and environment[15], an Open Source implementation of the S language developed at Bell Laboratories, is a natural environment for this type of simulation. It provides not only a high-level, interpreted programming language, but a rich collection of modern statistical and graphical methodology, and amongst other features, the facility to interactively monitor simulations as they progress. Beyond the specifics of the model discussed herein, we have developed a computational framework which can be readily extended to more complex and realistic models with little effort, and into which we can embed conventional network traffic simulators.
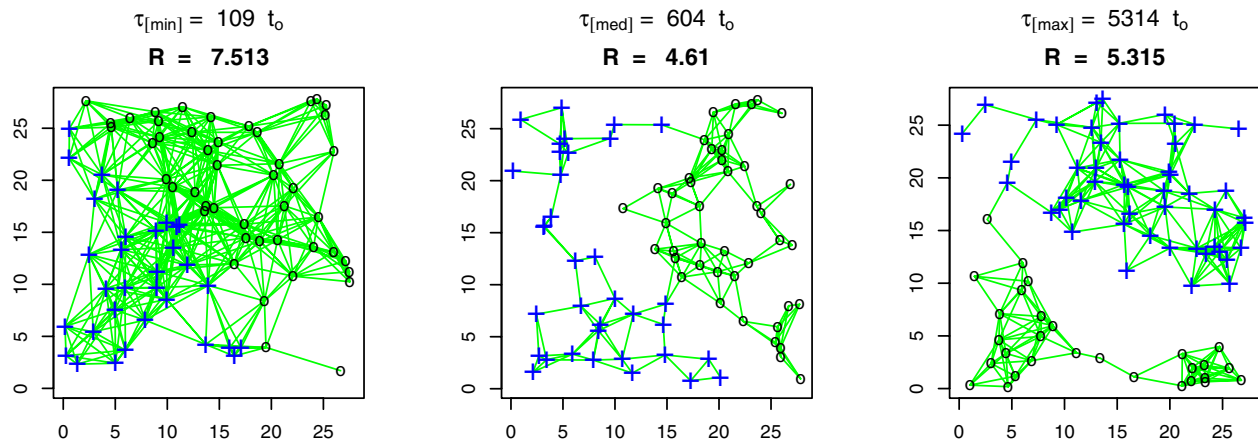
Fig. 5. Of the $n = 1000$ realizations of networks with $N = 78$ nodes distributed with spatial density $\rho = 0.1$, we illustrate the realizations with the minimum, the median, and the maximum relaxation times. These examples are representative of the common topology amongst realizations with short, average, and long relaxation times. Superimposed on the networks are markers indicating the initial condition with the longest relaxation time on that network, $\vec{v}_p$ (the eigenvector corresponding to λ_p). Nodes with a positive component in $\vec{v}_p$ are marked with pluses. Those with negative component are marked with circles.

D. Extremal and median behavior

As mentioned in Sec. II, we generated $n = 1000$ realizations of networks for various values of ρ and N. We focus on the realizations with $N = 78$ nodes uniformly distributed at random with spatial density $\rho = 0.1$, and those with $N = 80$ nodes and $\rho = 0.2$. Results for both densities are very similar, so only those for the former are shown explicitly. The connectivity requirement, R_c, for the $\rho = 0.1$ realizations constitute the histogram shown in Fig. 2. Our approach involves solving for the eigenvalues of a $N \times N$ matrix, so we focus on these realizations as they are for the largest values of N for which we could conveniently gather extensive statistics.

Once R_c is known we determine the penultimate eigenvalue λ_p for each realization, and hence the associated timescale $\tau = -1/\ln(\lambda_p)$. In Fig. 5 we single out three of the 1000 realizations for $\rho = 0.1$: the one with the shortest, the median, and the longest relaxation times. The significance of the plus signs and circles is explained in the subsequent section. The values of τ and of R_c are included above each realization. Note that we write the time in units of the discrete time increment t_o. In our simulations t_o corresponds to one complete synchronous update of the network (*i.e.*, each random walker is updated once). In an actual wireless network t_o is roughly the characteristic distance, R_c, divided by the data link speed.

The realizations shown in Fig. 5 are typical. Most of the networks with *short* relaxation times have a relatively large value for R_c, and hence are highly interconnected. Topologically, they tend to have a densely connected central region, and one or two nodes located at a large distance from any other node. Accommodating these outliers means R_c is larger than average and that nodes in locally dense regions broadcast at much higher power than necessary for minimal connectivity.

Networks with *long* relaxation times are not directly cor-
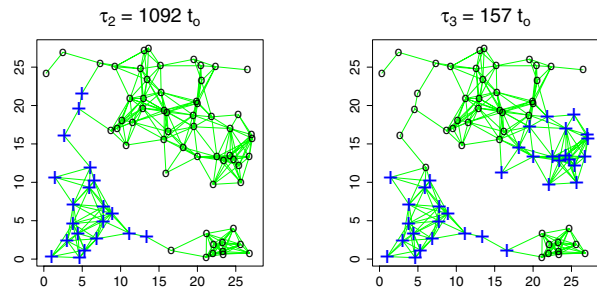

Fig. 6. Higher order modes for the realization with the maximum relaxation time, Fig. 5(c). Again the signs of the components in the corresponding eigenvalue are shown by the circles and pluses.

related with either large or small R_c, having instead a range of values. However, most of them have a topology similar to that shown in Fig. 5(c): two main connected sub-clusters with little cross-connectivity between them.

E. Simple bottlenecks and higher order modes

The eigenvalue, λ_p, determines the longest timescale. We can also gain information from $\vec{v}_p$, the associated eigenvector. The eigenvector $\vec{v}_p$ describes the initial condition for the amplitudes at each node with the longest relaxation time, typically having two distinct regions, one of positive and the other of negative components. In Fig. 5 we overlay on each node a marker indicating whether its amplitude given in $\vec{v}_p$ is negative or positive. Nodes with negative amplitude are shown by the circles. Those with positive amplitude are shown by the plus signs. As mentioned earlier, we can think of the circles and pluses as corresponding to two different viscous fluids diffusing on the network, and the relaxation time as the time required for the fluids to mix. The nodes located

at the transition between the regions of positive and negative correspond roughly to the bottlenecks for diffusion.

We consider also higher order modes (those with shorter relaxation times). For illustrative purposes we single out the realization shown in Fig. 5(c), and show, in Fig. 6, the mode associated with the next two largest eigenvalues. We label the timescale associated with each of these modes respectively as τ_2 and τ_3, and indicate the value above the corresponding figure. For each timescale, the network divides into sub-clusters. Nodes within each sub-cluster would be able to communicate within a time bounded by the corresponding timescale. This gives some indication of the range of communication accessible within that timescale and the size with which to build a routing table for that particular realization.

F. Distribution of timescales

By looking at individual instances we can gain insight into routing on particular networks. But we are more interested in general principles for routing on ad hoc networks with time-varying topologies, and ultimately in ad hoc networks made of *mobile* nodes. Here we are trying to establish limits in which we can treat the nodes as stationary, and moreover understand if behaviors of individual networks are similar to the average behavior.

We want to connect our study to one of networks with topologies that change in time. An important question to ask is to what extent does the longest timescale for diffusion of information vary over different realizations of networks? If the variance is large, and we find several instances with τ approaching infinite, building ad hoc networks of mobile elements will not be very feasible: as the nodes move, the longest timescale may jump from finite to near infinite, meaning we may have to wait close to an infinite amount of time to receive our data! One of the few previous studies of mobility in ad hoc networks also relies on the diffusion of data[3]. They show that the throughput of a network increases if the nodes are mobile, yet they do not address this issue of timescales or the distribution of timescales, both of which are critical for determining the feasibility of their scheme.

To understand the variance of timescales over independent realizations of networks, for every realization discussed thus far (those contributing to the histogram in Fig. 2), we determine the longest relaxation time, $\tau = -1/\ln(\lambda_p)$. In Fig. 7 we show the distribution for the 1000 samples with $\rho = 0.1$. (Results for $\rho = 0.2$ are very similar). Note we are recording the longest timescale for each network at the value of R_c for that network, hence many different values of R_c contribute to these plots. We are particularly interested in the upper tail of this distribution, understanding how frequently we should expect large outliers to occur. The density appears to decay exponentially, and we overlay on it an exponential density, $p(x) = \nu e^{-\nu x}$, where we set $\nu = \langle \tau \rangle$ (*i.e.*, we use the empirically obtained value of the average time, $\langle \tau \rangle$, as the parameter for the exponential density). Alongside the histogram, we show an exponential-quantile plot[16] comparing the exponential density and the empirical density. The
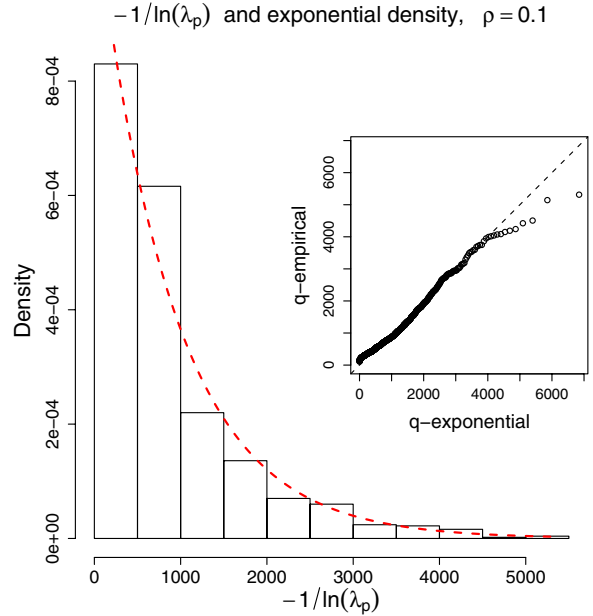

Fig. 7. Histogram of the empirical distribution of longest timescale, λ_p, over the $n = 1000$ independent realizations, for $\rho = 0.1$. The dotted line overlaying the histogram is the exponential density $\nu e^{-\nu x}$, where ν is determined by the empirical data, $\nu = \langle -1/\ln(\lambda_p) \rangle$. The accuracy with which the exponential distribution describes the empirical one is shown in the inlayed exponential-quantile plot.

values of the quantiles for the exponential density are plotted against the empirically determined quantiles. The dotted line with slope of unity describes the situation of exact agreement. This plot illustrates that the empirical distribution is accurately described by the corresponding exponential one. The departure in the tail (*i.e.*, the highest values) is due to the difficulty in estimating tail probabilities, and is well within the range of sampling variation, and accounts for less than one-percent of all the $n = 1000$ realizations studied.

Since the empirical distribution is well described by an exponential one, the moments of the distribution are small and finite. It would be extremely unlikely to observe an instance of a network with a value of $\tau \gg \langle \tau \rangle$. Furthermore this means the notion of a timescale for a specified ρ is a well-defined quantity, and as nodes move in space we do not expect major changes in the value of τ provided that ρ, the user density, does not change considerably.

IV. DISTRIBUTED ALGORITHM FOR NETWORK CONSTRUCTION

Up to now, we have been considering a scenario where all devices operate at the same power level and hence have the same transmission range, R_c. Though this assumption has advantages and approaches optimal in the asymptotic limit[8], it is not necessarily an efficient model to implement in practice. As is the case with sensor networks, optimizing for power usage may be the most critical factor. For other applications, it may be optimizing the topology to increase

throughput. We ultimately want to build up network topology in a distributed manner, using a local algorithm that, when compared with the CP algorithm, reduces the average per node power requirement, and increases the efficiency of the topology.

For processors distributed at random, in practice the spatial density of processors varies locally. Each realization has dense patches of nodes and other patches with just one or two nodes, as illustrated in Fig. 5. Note that nodes in dense regions end up over-connected. They could operate at much lower power, be connected more sparsely, and yet still be connected to the entire remainder of the network. The sparser connectivity would also result in less interference between simultaneous transmissions. The adaptive power algorithm discussed in [6] is a distributed construction for locally setting power to be the minimum necessary at each node, while still ensuring full network connectivity. However, instead of optimizing with respect to power usage as in [6], we want to optimize the efficiency of the topology with respect to τ, the longest timescale. We actually want some nodes to broadcast at higher power than necessary and hence have more connections than the minimum necessary. If certain nodes on the edges of the cluster broadcast at higher power, previously disjoint sub-clusters would connect up, allowing new paths through the network, which can introduce shortcuts and eliminate bottlenecks, resulting in more efficient network topologies.

See [8] for a brief review of existing approaches for topology control via distributed adaptive power constructions. We describe our adaptive power algorithm, then evaluate its performance relative to the common power level scheme using the metrics defined thus far. We show that this adaptive approach reduces interference and power consumption, and generates networks with more efficient topologies. In addition to these benefits, since the nodes set their power levels using information of the local environment, they can adjust these levels dynamically in response to changes in the local environment. In contrast, the common power level is not as robust to changes: if the nodes moved in space the value of R_c would vary. Each new value of R_c would first have to be determined and then broadcast across the system.

Building a fully connected network using only local information requires more than just specifying a required number of connections (as proposed in [17]), or using maximum nearest neighbor distance. Our scheme relies upon directional information, and is similar to [6]. It hinges upon the observation that nodes in locally dense regions tend to have connections distributed isotropically in all directions of space, yet nodes on the perimeter of clusters tend to have connections emanating from a small convex hull of connectivity, as can be seen in the networks illustrated in Fig. 5.

The adaptive power construction proceeds for each node independently. Each initially isolated node begins by transmitting at low power, gradually ramping up until either satisfying a geometric constraint on connectivity, as described below and illustrated in Fig. 8, or reaching a prespecified maximum allowed power level. As the node ramps up power it first
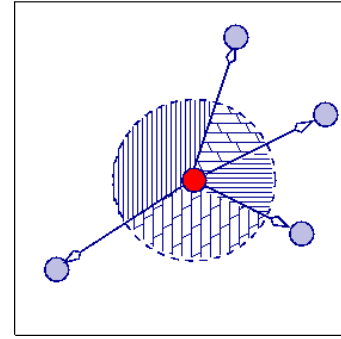

Fig. 8. The geometric constraint. If a node is connected to m neighbors, the vectors from the central node to the m neighbors divide a unit circle around the central node into m disjoint sectors. If the angle of each sector is less than or equal to π, the constraint is satisfied.

establishes a link with its closest neighbor, then with its next nearest neighbor (provided neither of these neighboring nodes is located further away than the maximum allowed range). With each new connection made, the geometric information is assessed. In general, when a node is connected to m neighbors, the vectors from the central node to the m neighbors divide a unit circle around the central node into m disjoint sectors. If the angle of each sector is less than or equal to π, the constraint is satisfied and the node sets its operating power at the current value. If any angle is greater than π the construction continues until either the central node makes a new connection, at which point the sector angles would be retabulated and the constraint rechecked, or the maximum operating power level is reached (in which case the power is lowered to the level where the last connection was established). The construction in [6] is very similar. However they use the value $2\pi/3$, which we believe is too conservative for our realizations. In addition, we explicitly set the maximum allowed operating power to be higher than the minimum necessary for connectivity.

As the construction progresses we build a connectivity matrix, M, as defined in Sec. III-A. Since each node constructs its connectivity independently of the other nodes, we occasionally give up reciprocity and construct uni-directional links. Thus the matrix M is not symmetric. For a discussion of the benefits and tradeoffs involved with uni-directional links see [18], and for a protocol level abstraction which deals with them see [19]. In the future we plan to compare current topologies to those which result when only bidirectional links are accepted.

The maximum range for connectivity was set to be $1.35 \times R_c$ (where R_c for each realization was the one determined by the CP construction). This range was chosen as it was found to be the smallest possible range for which every realization resulted in a fully connected network. Note we needed to ensure connectivity of all nodes for both transmission and reception of messages. This was confirmed by defining two connectivity matrices, one for transmission, M^T and one for reception M^R as follows,

$$M_{i,j}^T = M_{i,j} \quad \forall i > j \quad ; \quad M_{i,j}^T = M_{j,i} \quad \forall i < j;$$
$$M_{i,j}^R = M_{j,i} \quad \forall i > j \quad ; \quad M_{i,j}^R = M_{i,j} \quad \forall i < j. \quad (8)$$
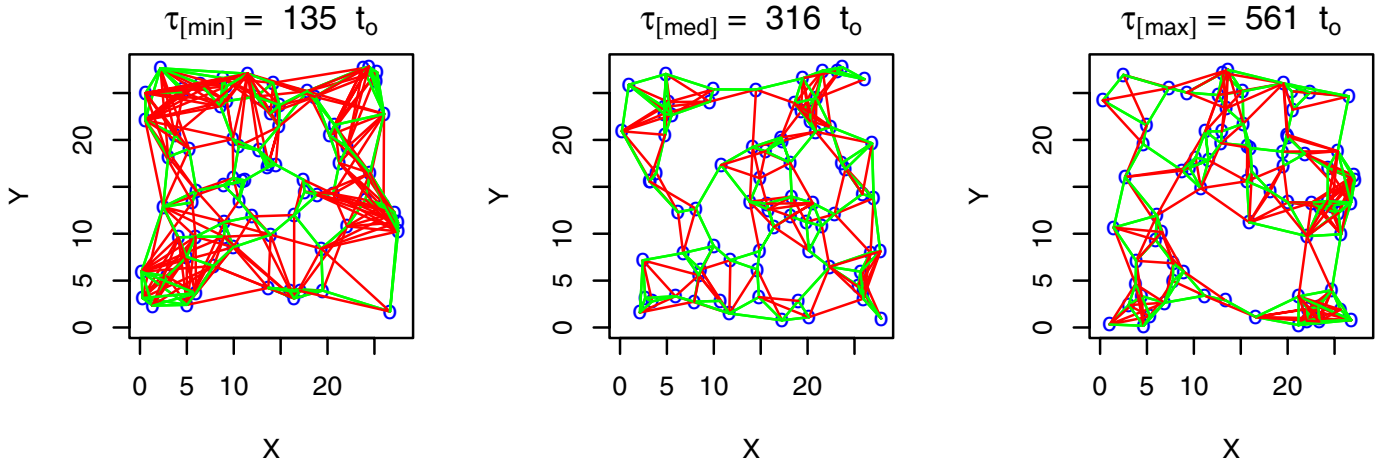
Fig. 9. Networks resulting from the AP construction. The node locations are identical to those for the realizations shown in Fig. 5. However the network topology results from the AP instead of the CP construction. The darkly drawn links are bi-directional, the lightly drawn ones are unidirectional. Note the more balanced load sharing, and the connections between sub-clusters that were almost disjoint with the CP construction. Such connections are "shortcut" paths through the network.

In other words, M^T is the lower triangular portion of the matrix M, mirrored across the diagonal. M^R is the upper triangular portion of M mirrored across the diagonal. Using these two connectivity matrices, we can define the state transmission matrices, P^T and P^R, as before and check that each of these have only a single eigenvalue that equals one, ensuring the network is composed of one fully connected cluster—thus we have a test for connectivity. Note, this test requires global information (the full connectivity matrix).

In Fig. 9, we show the networks which resulted for nodes with the same locations as shown in Fig 5, yet constructed with the AP instead of the CP scheme. The darkly drawn links are bi-directional, the lightly drawn ones are uni-directional. Note the AP constructions in general have more balanced load sharing, and many "shortcut" paths connecting together sub-clusters that were almost disjoint with the CP construction.

We can quantitatively compare the alternate constructions directly with two different performance metrics, the first based on timescales, the second based on expected power consumption. We do this for all $n = 1000$ instances with $\rho = 0.1$ discussed thus far (note, as with all other results presented, the comparisons for the instances with $\rho = 0.2$ are extremely similar, so we choose not to reproduce them here). Fig. 10 shows two different comparisons of the timescales. Fig. 10(a) is a histogram of the values of τ which result using the AP as opposed to the CP construction. The overlayed dashed line is the same exponential density plotted in Fig. 7, which accurately described the envelope of the histogram for the CP construction. Note the significant shift toward shorter timescales, which reflects not only the "shortcut" paths, but also reduced interference. Fig. 10(b) is a scatterplot directly comparing τ for each individual realization generated first by the CP and then the AP construction. In almost every instance

except one, the AP timescale was considerably shorter. And note that the extreme cases, with the largest values of τ under the CP construction, have small values of τ with the AP. So for these extreme instances the adaptive algorithm is especially superior under this measure.

We also compare the relative power consumption of the networks generated with the alternate constructions. We assume that all nodes are transmitting all the time. Furthermore, as a rule of thumb, we assume that the power falls off with distance R as, $P \propto 1/R^{2.5}$. Since we know the value of the transmission range for each node, we can thus calculate its power consumption. In Fig. 11 we show a scatterplot comparing average power consumption for the 1000 instances with $\rho = 0.1$. The horizontal axis denotes the power consumption with the CP scheme. The vertical axis denotes power consumption with the AP scheme. The slope, $\delta = 0.41$. Thus for each unit of power increase with the CP scheme, we expect only a 0.41 unit increase with the AP scheme.

V. DISCUSSION AND CONCLUSIONS

In this manuscript we have attempted to accurately define the performance metrics relevant to ad hoc networks. They are power consumption, interference, and the characteristic time for message delivery. We also discuss an adaptive power algorithm for network construction, and using these metrics, assess its performance relative to a more standard algorithm.

The characteristic time, τ, required for a message to be delivered via peer-to-peer communication in an ad hoc network is a fundamental quantity. Not only does this timescale potentially constrain the feasibility of building such networks, it gives an estimate of the time required to build up a complete routing table for the network. Furthermore it serves as a
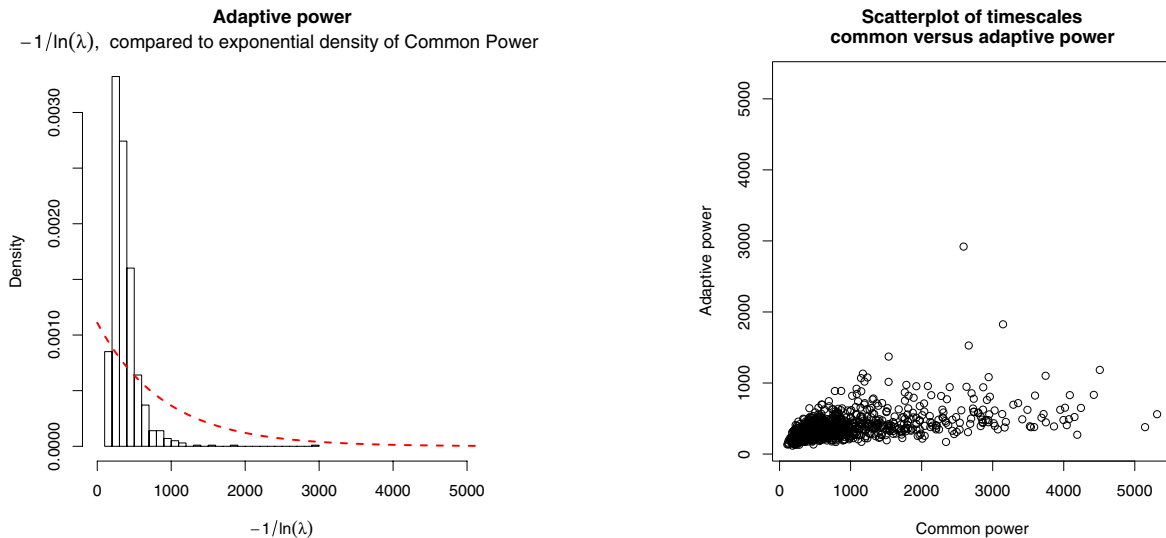
Adaptive power
$-1/\ln(\lambda)$, compared to exponential density of Common Power

Scatterplot of timescales
common versus adaptive power

Fig. 10. (a) A histogram of the values for τ for the same $n = 1000$ realizations with $\rho = 0.1$ discussed thus far, however constructed with the AP instead of CP schemes. The dashed line is the same exponential density plotted in Fig. 7. (b) A scatterplot comparing each of the 1000 instances. The horizontal axis denotes the value of τ resulting from a CP construction, the vertical axis, from the AP construction. There is one obvious outlier where the value of τ was not lowered by the AP construction. In all other case τ is lowered, which is especially significant for those realizations with the largest values of τ under the CP construction.

performance metric for evaluating alternate network construction schemes. Yet, despite its relevance, no previous work has quantitatively discussed the relevance nor attempted to quantify the characteristic time. We introduce a framework based on the assumption that messages diffuse along the network. Diffusion means no strategy is used to efficiently exchange data. If any strategy for routing messages were used we would expect the value of τ to decrease. Thus the time obtained by our method is an upper bound on the actual time.

We are also interested in the distribution of this time across many independent realizations of networks with similar user densities. This distribution gives insight into the feasibility of communicating efficiently with ad hoc networks with time-varying topologies. In particular, if the distribution has a large variance, we would expect the time to exhibit large fluctuations as the underlying network topology changes. Instead we find the empirical distribution is well described by an exponential distribution. Hence the fluctuations on average will not be large and the timescale will not change drastically if the underlying topology changes while the local user density remains fixed.

We also introduce a decentralized algorithm for network construction, which is a variant of [6]. Instead of optimizing with respect to minimal power, we optimize with respect to minimizing τ, the largest timescale. Our adaptive power (AP) algorithm lets each device set its power level individually to optimize its own connectivity, using only information of the current state of its local environment. When compared to networks generated with the standard common power (CP) algorithm the networks resulting from the AP construction have more efficient network topologies and improved performance by all three metrics (power consumption, interference,

and timescales). In addition, since the AP scheme uses only information local to each device, the topology of the network can change rapidly in response to environmental changes, such as moving users or time-varying wireless channels; the construction can be iterated locally as necessary. In fact many routing algorithms rely upon continually executing route discovery algorithms, such as ad hoc on demand distance vector routing[7]. Using the AP construction for topology and route discovery, "shortcut" paths through the network are found which would not be found with the CP approach. Thus the AP networks should have a higher throughput than the CP ones.

With regards to the power consumption metric, we assumed that all devices were transmitting at all times (*i.e.*, a high traffic density limit). More accurately we could instead estimate usage and use this to determine power consumption. In addition, past algorithms for adaptive power during usage can be overlayed onto the networks we constructed. Such algorithms are based on devices transmitting at the minimum power necessary to reach a specific neighbor, not the minimum power for full network connectivity[20].

Much future work which fits naturally into our framework involves quantifying distinct regimes for routing. For instance, answering the question on how large to build routing tables and how often to refresh them. We expect such answers to depend on a combination of factors described herein, such as the density of nodes and traffic, and the relative speed of the devices with regards to the instantaneous value of the characteristic time. We now have a way to quantify the characteristic time. Also, depending on the distance between the source and destination, we expect that different strategies

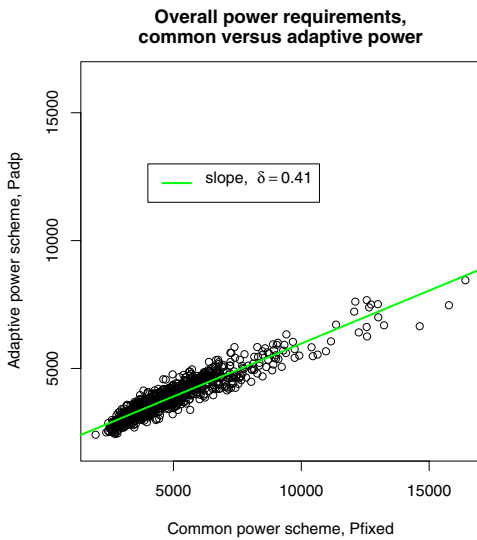## Overall power requirements, common versus adaptive power

Fig. 11. A scatterplot comparing average power consumption for the 1000 instances, where we assume all nodes are transmitting at all times, and that the power decays as a function of distance R as $P \propto 1/R^{2.5}$. The horizontal axis denotes the power consumption with the CP scheme. The vertical axis denotes power consumption with the AP scheme. The slope, $\delta = 0.41$. Thus for each unit of power increase with the CP scheme, we expect only a 0.41 unit increase with the AP scheme.

for routing would be better. Consider the network show in Fig. 5(c) and Fig. 6. We have illustrated the sub-clustering that occurs for different timescales. Clearly, based on timescale constraints, different routing strategies would be necessary for efficient intra-cluster and inter-cluster communication.

There are many challenges still not addressed in ad hoc networking. As shown recently, the assumption intrinsic to almost all existing models of wireless network topology, that received power falls off monotonically with distance from the source, is not necessarily valid[21]. We are currently developing simple models to generate realistic wireless footprints which can then be coupled to the CP and AP models. Routing strategies need to be specifically tailored to the underlying network building algorithm. Our adaptive power algorithm would require that the devices have either directional antennae or other means of directional sensing[22]. Design issues for such systems need to be considered. Ad hoc networks also provide us with opportunities for intelligent noise cancellation schemes, which have yet to be introduced and studied. Furthermore, actual ad hoc networks will experience noise and scattering from the environment which need to be understood. Finally, recent work has focused on hybrid networks with some ad hoc and some base-station communication, along with heterogeneous clients[23]. As mentioned in Sec. I, quantifying timescales for ad hoc networks with heterogeneous clients is particularly pertinent.

REFERENCES

[1] *Embedded, Everywhere: A Research Agenda for Networked Systems of Embedded Computers.* Computer Science and Telecommunications Board (CSTB) Report, 2001.
[2] T. S. Rappaport, *Wireless communications : principles and practice.* Upper Saddle River, N.J.: Prentice Hall, 1996.
[3] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad-hoc wireless networks," in *Proceedings of INFOCOM*, 2001.
[4] E. N. Gilbert, "Random plane networks," *J. SIAM*, vol. 9, no. 4, pp. 533–543, 1961.
[5] P. Gupta and P. R. Kumar, "Critical power for asymptotic connectivity in wireless networks," in *Stochastic Analysis, Control, Optimization and Applications*, W. M. McEneaney, G. Yin, and Q. Zhang, Eds. Boston: Birkhäuser, 1998.
[6] R. Wattenhofer, L. Li, P. Bahl, and Y. Wang, "Distributed topology control for power efficient operation in multihop wireless ad hoc networks," in *Proceedings of IEEE INFOCOM*, 2001.
[7] C. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing," in *MILCOM '97 panel on Ad Hoc Networks*, 1997. [Online]. Available: citeseer.nj.nec.com/136139.html
[8] S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. R. Kumar, "Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the COMPOW protocol," in *Proceedings of European Wireless Conference*, 2002.
[9] M. D. Penrose, "A strong law for the longest edge of the minimal spanning tree," *Ann. Probab.*, vol. 27, no. 1, pp. 246–260, 1999.
[10] P. Karn, "MACA - a new channel access method for packet radio," in *AARU/CRRL Amateur Radio 9th Computer Networking Conference*, 1990, pp. 134–140.
[11] M. Liljenstam and R. Ayani, "Interference radius in PCS radio resource management simulations," in *Proceedings of the 1998 Winter Simulation Conference*, 1998, pp. 1629–1637.
[12] K. Ramanan, A. Sengupta, I. Ziedins, and P. Mitra, "Markov random field models of multicasting in tree networks," To appear in *J. Appl. Prob.*
[13] G. Zysman et. al., "Technology evolution for mobile and personal communications," *Bell Labs Technical Journal*, pp. 107–129, January–March, 2000.
[14] J. R. Norris, *Markov Chains*. Cambridge, U.K.: Cambridge University Press, 1997.
[15] R. Ihaka and R. Gentleman, "R: A language for data analysis and graphics," *Journal of Computational and Graphical Statistics*, vol. 5, no. 3, pp. 299–314, 1996.
[16] S. Kotz, N. L. Johnson, and C. B. Read, Eds., *Encyclopedia of Statistical Sciences*. Wiley, 1982.
[17] R. Ramanathan and R. Rosales-Hain, "Topology control of multihop wireless networks using transmit power adjustment," in *Proceedings of IEEE INFOCOM*, 2000, pp. 404–413.
[18] M. Marina and S. Das, "Routing performance in the presence of unidirectional links in multihop wireless networks," in *Proceedings of MOBIHOC*, June 2002.
[19] V. Ramasubramanian, R. Chandra, and D. Mossé, "SRL: Providing a bidirectional abstraction for unidirectional adhoc networks," in *Proceedings of IEEE INFOCOM*, June 2002.
[20] B. Narendran, J. Sienicki, S. Yajnik, and P. Agrawal, "Evaluation of and adaptive power and error control algorithm for wireless systems," in *IEEE International Conference on Communications (ICC'97)*, 1997.
[21] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker, "Complex behavior at scale: An experimental study of low-power wireless sensor networks," *http://citeseer.nj.nec.com/533751.html*.
[22] L. Girod and D. Estrin, "Robust range estimation using acoustic and multimodal sensing," in *Proceedings of the IEEE/RSJ Intl Conf on Intelligent Robots and Systems (IROS)*, 2001.
[23] T. Phan, G. Zorpas, and R. Bagrodia, "An extensible and scalable content adaptation pipeline architecture to support heterogeneous clients," To appear at *The 22nd International Conference on Distributed Computing Systems*, July 2-5, 2002, and P. Gupta, private communication.