Threshold-based Policies in Mobile Infostation Networks

Furuzan Atay & Christopher Rose
WINLAB, Rutgers University
73 Brett Rd.,Piscataway, NJ 08854-8060
Email: furuzan/crose@winlab.rutgers.edu

***Abstract*— Mobility can increase throughput in ad hoc networks by providing channel variation. If delay constraints are very loose, it is possible for a given packet to observe many different network topologies as nodes move relative one another, and these different topologies can be treated as diversity. Opportunistic strategies can exploit large scale changes in the channel quality to achieve higher throughput. However, to attain these higher capacities, delays on the order of the node mobility time constants must be tolerated.**

In this paper, we design cost functions based on delay and transmit power and implement simple greedy cost-minimizing strategies to enable the trade-off between mobility, delay and throughput. In particular, we study the performance of networks where all the packets are routed using simple threshold rules. We also examine the scaling properties of throughput and delay of our strategies.

I. Introduction

Recently, it has been shown that for networks of geographically fixed nodes throughput capacity is ”not scalable”. That is, in the limit of large numbers of nodes, throughput capacity per node goes to zero. Even if transmission ranges and transmission schedules are chosen optimally, as the number of nodes increases, throughput capacity per node decreases as $1/\sqrt{N}$ where N is the number of nodes. [1]

One new line of thought involves node mobility – previously seen as undesirable since it complicates routing and can cause packet loss owing to intermittent node connectivity. However, if packets have loose delay constraints, mobility can not only increase throughput but also make capacity per user scalable [2]. The key idea is limiting both the number of hops [1] and the average hop length simultaneously regardless of network size – an impossibility in any fixed network under uniform traffic assumptions. This new idea constitutes the theoretical motivation behind the ”mobile infostation architecture” [3].

In mobile infostations a packet can travel between nodes when the conditions are favorable and sojourn at a relay node in the meantime. That is, relay nodes need not forward a packet as soon as it is received. If the next hop is too costly (in terms of some suitable network resource), packets can be retained until the next good transmission opportunity. It is possible for a given packet to observe many different network topologies as nodes move relative one another, and these different topologies can be treated as diversity. In generic ad hoc networks, many applications have strict delay constraints requiring the network to be connected most of the time. A mobile infostation architecture, on the other hand, targets applications with loose delay constraints and high data rate requirements. Thus, intermittent connectivity is both tolerated and expected.

[1]In the case of [2], the number of hops was limited to two.

In this paper, we study the trade-off between throughput and delay in mobile infostations. We define cost functions based on transmit power and delay, and implement simple greedy optimizations at packet level. In [4], different packet-oriented strategies have been studied and it has been concluded that from a practical and computational perspective, threshold rules are the best candidates to be used in network studies in which packets interact. Thus, we base our protocol on threshold rules.

II. Related Work

In [3], the authors studied the throughput performance of mobile infostations by accepting a potentially large delay. The optimal transmission ranges to maximize local throughput is calculated and it was shown that the optimal transmission range of mobile infostations is much shorter than (5 to 10 times) that of generic ad hoc networks.

Lately, there have been attempts to improve the delays of architectures that rely on mobility [5]–[7]. One way of decreasing the packet delay is generating and distributing more than one copy of the same packet [5] and [7]. Another way is to allow each packet to make more hops than required[2]. [6] considers a multiple hop approach to alleviate delay. However, the proposed protocol requires a two-tier hierarchical architecture where the sources and destinations are stable and the relays are mobile. It also assumes that all mobile nodes know their future trajectories for random time and they share this knowledge with each other and stable nodes.

In this work, we consider a homogenous network where all the nodes are mobile. We assume that nodes can obtain updated topology information (not necessarily global topology) after each hop, but does not know future topologies.

The rest of this paper is organized as follows: First, we will introduce the basic model and the cost structure. In section-IV we will describe the threshold based packet policy. A multiple packet simulation model will be described in section-V. In section-VI we will examine the results.

III. Model and Assumptions

We consider a network of packets that interact and compete with each other for network resources. The objective is to achieve low delay and high throughput. We define a cost function reflecting these objectives and evaluate optimal operating points by varying the weight of cost components.

We assume discrete intervals of duration δ during which packets can move directly between two nodes or stay put to await more favorable conditions. Time is measured in integer

[2]The minimum number of hops required to achieve a scalable throughput depends on the system details like the traffic model. In the case of [2] at least 2 hops are required.

units of δ. The cost, $c_{ij}(t) = c_{ji}(t)$, of transmissions between nodes i and j is a function of time owing to node mobility. Every packet has a unique destination. We assume that each packet can obtain updated topology information after each hop, but does not know future topologies.

In our mobility model, N independent nodes constrained to the plane move in a Brownian fashion. To avoid boundary effects, both x- and y-axes are wrapped around forming a torus. For all experiments we choose $\delta = 0.010\bar{a}^2/D$ where $\bar{a}$ is the average internodal distance and D is the diffusion coefficient. This number is chosen such that in δ seconds significant channel variations occur owing to mobility.

A. Cost Structure

Our cost structure has two components: delay cost and "social" cost. The delay cost is simply the time required for the packet to traverse a given link. The social cost accounts for the interference among the packets. It is defined as the average number of other nodes which are affected by the interference associated with the transmission from node i to node j.

Achieving a target signal to interference/noise ratio (SINR), γ^*, at the receiver is assumed sufficient for successful transmission at some fixed rate R. The received power $P_j^{(r)}$ at node j due to a transmission from node i with power P_i is

$$P_j^{(r)} = P_i \left(\frac{d_{ij}}{d_0} \right)^{\alpha} \tag{1}$$

where d_{ij} is the distance from node i to j, $d_0 \leq d_{ij}$ is some minimum distance, and α is the propagation exponent. Accordingly, in the absence of interference, the minimum transmit power required for successful transmission is given by:

$$P_{ij}^* = N_0 W \gamma^* \left(\frac{d_{ij}}{d_0} \right)^{\alpha} \tag{2}$$

where W is the available bandwidth and N_0 is the background noise spectral intensity. We assume that the transmitter cannot transmit with arbitrarily small power and the minimum possible power level is equal to P_{ij}^* of distance $d_{ij} = d_0$.

If the target SINR is γ^* and the ambient noise level is $N_0 W$ then we define the *interference radius* R^* as that distance within which the interference produced by the transmission at power P_i results in interference of magnitude $\beta N_0 W$ where $0 \leq \beta \leq 1$. Thus, we have

$$\beta N_0 W = P_i / \left(\frac{R^*}{d_0} \right)^{\alpha} \tag{3}$$

which results in an interference radius

$$R^* = d_0 \left(\frac{P_i}{\beta N_0 W} \right)^{1/\alpha} \tag{4}$$

and an associated social cost of

$$C_S = \rho \pi d_0^2 \left(\frac{P_i}{\beta N_0 W} \right)^{2/\alpha} \tag{5}$$

where ρ is the node density.

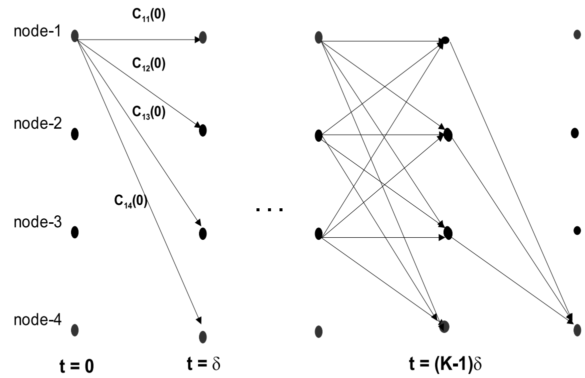

Fig. 1. A graph depicting all possible tours through a network of four nodes assuming packet origination at node 1 and forced termination by time $t = T$ at node 4. Link costs are shown for the first hop. Since 4 is the destination node, it is always a terminal node wherever it appears in any tour.

In a multiuser environment SINR at node-j for a transmission from node-i is given by:

$$\gamma_{ij} = \frac{P_{ij}}{\sum_{k=0, k \neq i}^{N} P_{kj} + N_0 W} \tag{6}$$

where P_{ij} is the power received at node-j due to a transmission at node-i. For successful reception, nodes have to transmit with a power larger than the value given in equation (2). Thus, we assume that transmitting nodes choose their power levels by targeting a higher SNR than the actual target required for successful transmission:

$$P_{ij}^* = N_0 W \tilde{\gamma} \left(\frac{d_{ij}}{d_0} \right)^{\alpha} \tag{7}$$

where $\tilde{\gamma} > \gamma^*$.

Using equation (7) in equation (5) we obtain:

$$C_S = \rho \pi \left(\frac{\gamma^*}{\beta} \right)^{2/\alpha} d_{ij}^2(t) \tag{8}$$

Thus, the total cost for a transmission from node i to node j can be defined as

$$c_{ij}(t) = w_d \delta + w_s \rho \pi \left(\frac{\gamma^*}{\beta} \right)^{2/\alpha} d_{ij}^2(t) \tag{9}$$

where w_d and w_s are positive weighting constants assigned to delay and social cost respectively.

Given link costs, packet motion from a source node, σ, to a destination node, Δ, through the network can then be modeled as a graph such as that depicted in FIGURE 1. Note that d_{ii} is assumed zero. The cost of a given tour is the sum of costs for the links traversed. Formally, if we denote a tour $\mathcal{T}$ by a sequence of integers $I_{\mathcal{T}} \equiv \{i_1, i_2, \cdots, i_K\}$ then

$$C_{\mathcal{T}} = \sum_{k=1}^{K} c_{i_{k-1} i_k}(k-1) \tag{10}$$

Here, the individual $c_{ij}(t)$ could be either deterministic or represent snapshots of a random process driven by the node mobility and other external time-varying processes.

On such a graph, the minimum cost of delivering a packet to its destination within time T is expressed as:

$$C = \min_{\mathcal{T}} C_{\mathcal{T}} \tag{11}$$

or for stochastic link costs

$$\bar{C} = \min_{\mathcal{T}} E\left[C_{\mathcal{T}}\right] \tag{12}$$

and optimal schedules

$$\mathcal{T}^* = \arg\min_{\mathcal{T}} C_{\mathcal{T}} \tag{13}$$

or again for stochastic costs

$$\mathcal{T}^* = \arg\min_{\mathcal{T}} E\left[C_{\mathcal{T}}\right] \tag{14}$$

In the deterministic case, standard dynamic programming (DP) methods can be used to calculate optimal schedules with $O(N^2K)$ computations where $K = T/\delta$ is the total number of available steps [8], [9]. When the node positions are random snapshots of the underlying independent Brownian motion and the packet can obtain updated topology information after each hop, link costs constitute a Markov process. The exact solutions to this problem are often complex [4]. Under the same assumptions, a threshold-based policy called "eager packet policy" has been proposed in [4]. It has been shown that this policy performs within some factor of the best possible performance which is achieved by the optimal policy under maximum topology knowledge. Throughout this paper, we use the "eager packet policy" to analyze the delay-throughput trade-off.

IV. Packet Routing Policy: Eager Packets

We assume that a packet is allowed to modify the tour based on new node location information obtained at each time step. At each step, it calculates the minimum cost of reaching to its destination by making at most k hops. For this calculation it assumes that the link costs will be fixed for the next k steps. If the calculated cost is smaller than some threshold, V_t, it moves all the way to its destination. Otherwise, it waits until the next step.

There are three parameters: the cost coefficients, (w_d, w_s), and the threshold level, V_t. Without loss of generality, we set $V_t = 1.0$ and vary the cost coefficients to obtain different possible delay-throughput points.

Note that w_d and w_s cannot be chosen arbitrarily. Due to the minimum transmit power constraint, the minimum possible tour between two nodes is a single hop of length d_0. Coefficient pairs used should result a total cost smaller than $V_t = 1.0$ for this minimum cost tour. Otherwise, tour completion is impossible. Let $c(d, w_d, w_s)$ denote the cost of a link of length d when the cost coefficients are w_d and w_s. We recall equation (9) which states that the link cost is of the following form:

$$c(d, w_d, w_s) = w_d c_1 + w_s c_2 d^2 \tag{15}$$

where c_1 and c_2 are positive numbers. Then, we can calculate the maximum values of w_d and w_s by solving:

$$c(d_0, 0, w_{p,max}) = 1.0 \quad c(d_0, w_{d,max}, 0) = 1.0 \tag{16}$$

We also see that the region of valid cost coefficients is a triangle due to the additive cost structure.

We note that it is possible to put a limit on the maximum number of hops the packet takes by choosing an appropriate w_d. For example, when $w_d > w_{d,max}/2$, the packet can never make more than one hop. If the number of hops is limited to one, the strategy becomes a threshold rule on the transmission distance where threshold distance, d_t, corresponding to a particular (w_d, w_s) is obtained by:

$$c(d_t, w_d, w_s) = 1.0 \tag{17}$$

Moreover, in one-hop region the same d_t can be obtained with different coefficient pairs. From equation (15) and equation (17), it can be easily seen that cost coefficients (w_d, w_s) that correspond to the same d_t form a line defined by:

$$c_1 w_d + c_2 d_t^{\alpha} w_s - 1 = 0 \tag{18}$$

A. Eager Packet Algorithm

The most straight forward algorithm is the following: At every δ nodes move and at each node a trellis as in FIGURE 1 is formed. When the number of hops is limited to k, the trellis consists of N nodes and k steps. On such a trellis finding the minimum cost from a given source to all other nodes requires $0(N^2k)$ computations. The total number of computations depends on when the packet accepts a trellis it observed. Thus, as V_t decreases, the average of the total number of computations increases. When the threshold level V_t is low and the number of nodes is large, the computational complexity of this method might be too large.

A better method involves a forward DP algorithm which calculates costs from a common source to "some" of the other nodes. Instead of solving the DP at each epoch and comparing the total cost to a threshold we start with a given threshold. The algorithm correctly finds the minimum cost tour for destinations that can be reached with a total cost smaller than V_t. For the rest of the destinations, in which we are not interested due to high cost, it returns an incomplete tour and an arbitrary total cost larger than V_t.

The idea is while progressing forward in the DP, to store only those branches whose total cost so far are smaller than V_t. Other branches are "pruned" since they can never lead the packet to any of the destinations with cost smaller than the threshold. For small V_t most of the nodes are along the way to the last step reducing the number of computations.

B. Implementation of Threshold Policy

All the packets are assumed to use common cost coefficient pairs (w_d, w_s) and unit threshold, $V_t = 1$. The optimal (w_d, w_s) of a network is defined as the value at which average packet delay is minimized while achieving a throughput equal to the network load. When a collision occurs, the packet restarts the algorithm as if it has been generated by its current relay node. For simulation purposes, we assume that all the nodes have a global view of the network. However, the policy can be implemented as a distributed and scalable protocol.

C. *Scalability of Distributed Threshold Policy*

Routing information in a threshold policy is inherently scalable for any finite value of threshold, V_t. In our architecture, a node does not need to know about the paths to all other nodes but only a fraction of them. Each node is interested in only the nodes that are reachable with a cost smaller than V_t. For example, in a Distance Vector approach [10], routing messages can be modified to include the total cost. A node only sends the information that might be useful to its neighbor. The node considers an entry as useless to a neighbor if the total cost calculated by that neighbor would be larger than the threshold V_t, thus this information would actually never been used. In this way, the size of the routing messages are limited by the threshold. Although we do not analyze exactly how the number of nodes effects the message size, it is clear that this approach has much better scaling properties than the approaches for fixed ad hoc networks.

V. Simulation Model Description

A. *Performance Metrics*

Our performance metrics are average packet delay and throughput. Due to the ergodicity of all processes involved, we can measure the average delay and throughput by taking time averages. We define the throughput, which can also be called long-term throughput, in a way similar to [2]:

$$Tp = \frac{1}{NK}\sum_{t=0}^{K}\sum_{j=1}^{N} M_j(t) \tag{19}$$

where $M_j(t)$ is the number of packets successfully delivered to node-j at step t, N is the number of nodes and K is the total time in steps. Total simulation time must be long enough for the network to observe a variety of topologies. We characterize this time by $\tau_D = A/D$ where D is the diffusion coefficient of the Brownian motion and A is the area and choose $T = K\delta \geq 10\tau_D$.

B. *Traffic Model:*

A uniform traffic matrix is assumed. That is, every node tries to communicate with every other node with equal rates. The packet generation at a node is modeled as a Poisson process with rate λ. To have a uniform traffic, each packet is assigned to a destination that is chosen at random. λ is measured in packets/step/node.

C. *Packet and Node Contention*

As the number of packets in the network increases, conflicts arise among the intended tours of different packets. These conflicts are classified into two groups: *node-level contentions* and *packet level contentions*. Nearby nodes transmitting simultaneously interfere with each other causing some of the transmissions to fail. We define these conflicts as *node-level contention*. *Packet level contentions* are caused by packets buffered at the same node and scheduled to hop at the same step.

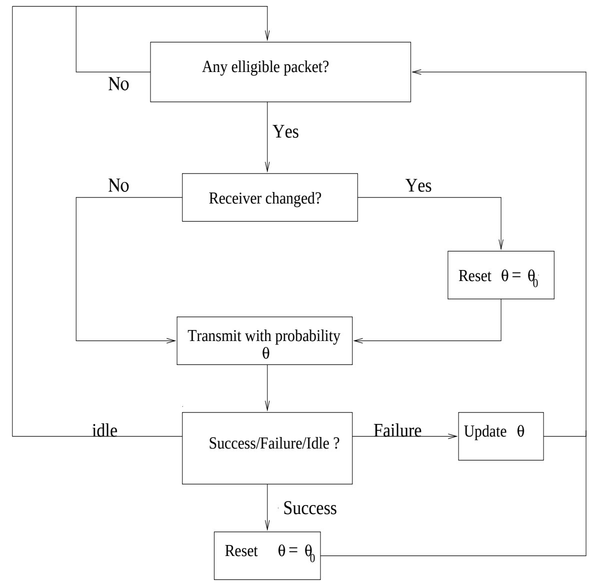

Fig. 2. Policy for node-level contentions

Coupling these two problems might improve the overall performance. For example, a packet at the end of its queue can be transmitted by bypassing the other packets in the queue if the node-level interaction assures that this transmission is more likely to be successful. However, this kind of coupling complicates the problem and will not be considered here. In our system, the packets to be sent are chosen independent of the current interference structure and then the nodes having packets to send compete with each other.

1) Node-level Contentions: To resolve conflicts among the nodes, we employ a slotted ALOHA based protocol. We assume that there exist an error-free side channel for acknowledgements and use a back-off algorithm to control the channel access in a more adaptive way. We use the algorithm shown in FIGURE 2. We choose $\theta_0 = 1.0$ and update θ using

$$\theta(n) = \theta(n-1)/r \tag{20}$$

where $\theta(n)$ denotes the transmission probability at time step n and r is called the back-off factor. In our simulation studies, as soon as its receiver changes, the node declares a topology change and set its $\theta = \theta_0$. In our simulation studies, we used $r = 2$ and observed that the algorithm is delay-stable almost all the time for stable loads.

2) Packet-level Contentions: At a given slot, each node decides which packets to transmit among all eligible packets, i.e, packets whose minimum cost to their destination is smaller than the threshold. Each node maintains a single queue of all the packets it carries and picks the first eligible packet in this queue. An alternative to the above strategy could be to give priority to the packets with smaller costs. However, this

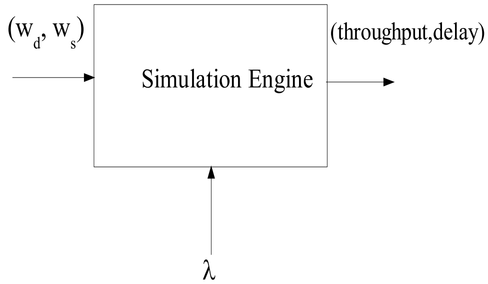

Fig. 3. Simulation set up: Input variables are the cost coefficients (w_d, w_s). Other basic parameter is the network packet generation rate, λ.

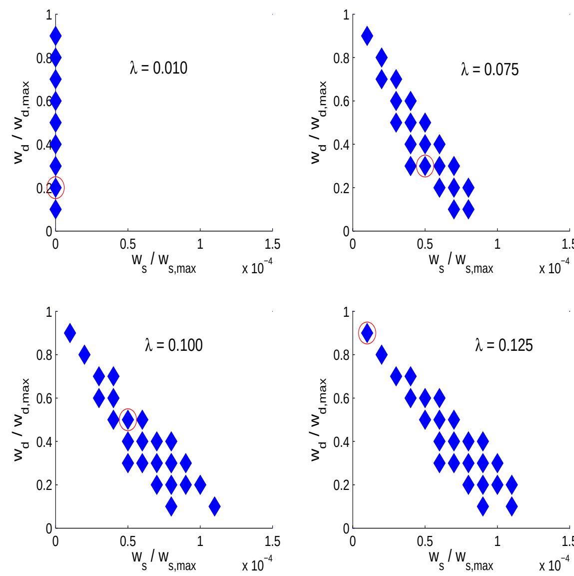

Fig. 4. Small delay regions for 4 different λ (in packets/node/step) Delay tolerance, μ is chosen as 20%. The minimum delay points are circled.

strategy requires more computation by the nodes and it is not considered here.

VI. Results

A. Minimizing Delay Through Cost Coefficients

Here, we find the regimes where the packet delay is minimized at different packet generation rates. First, we do not restrict the maximum number of hops, k, i.e. we set $k = \infty$, and exercise control only through the cost coefficients. FIGURE 3 illustrates the simulation set up. The variables are the cost coefficients, (w_d, w_s) and the network load, λ. We study a few representative load levels. The rest of the parameters are kept fixed. We have 15 nodes, $N = 15$, and the propagation constant is 4.0 ($\alpha = 4.0$). For a given λ, the problem is a 2-dimensional optimization where the objective function is not an analytical expression: it is a "black box" that consists of a simulation engine in which all the packets use a threshold rule with the given cost coefficients (w_d, w_s) and threshold value, $V_t = 1$. That makes the use of standard optimization algorithms impractical since all of them require the calculation of gradient function at every iteration [11]. So, we do an exhaustive search with discrete steps. We evaluate the delay and throughput at a wide range of (w_d, w_s) pairs. Instead of showing point(s) where the delay takes its minimum value, we mark the "small delay regions". Small delay region is defined as the area where the measured delay is within some percentage, μ, of its minimum value.

FIGURE 4 shows the small delay regions for four different λ values. Under light load, $\lambda = 0.01$, which is less than 10% of the maximum throughput, delay is minimized when $w_s = 0$. If $w_s = 0$, all hops have equal cost and thus, the minimum cost tour is a direct hop from the source to the destination. In this case, the exact value of w_d is irrelevant. For the other three load levels, we see that the small delay region moves to the left on the w_s axis as the load increases making the policy more interference-aware.

As in the isolated packet analysis [4], w_d upper bounds the number of hops and in the region where $w_s/w_{s,max} \geq 0.5$ the packets cannot make more than one hop. In the one-hop region, the policy can be characterized by a single parameter: threshold distance, d_t. In section-IV we argued that for each point, P, in one-hop region, there are other points that correspond to the same threshold distance as P and these points constitute a line in the $w_d - w_s$ plane. Accordingly, in FIGURE 4, at all load levels, the small delay region takes the shape of a line in the one-hop region.

An interesting observation is that in all four cases, the small delay region intersects with the one-hop region. That is, if a one-hop threshold policy is imposed instead of allowing unlimited number of hops, the delay can increase at most $\mu = 20$ percent. Under light load, direct hops are preferable because of their short delays. From [4] we know that when the cost coefficients are large, the paths taken are usually one-hop paths. Thus, under high loads (corresponding to larger (w_d, w_s)), no benefit is expected from multiple-hops. At load levels between the two, queuing delay at relay nodes is so large that the our fixed link costs assumption breaks. Then, once again, a one-hop policy emerges. We will look at one-hop threshold rules more closely in section-VI-B.

B. One-Hop Threshold Rule: Delay and Throughput

Now, we will look at the effect of node density on the performance of one-hop threshold policies. To control node density, we fix the area and vary the number of nodes, N. We note that for $k = 1$, choosing optimal (w_d, w_s) while fixing $V_t = 1$ is equivalent to choosing an optimal maximum transmission distance (d_t^*).

1) Scaling of the Delay: Now, we study the behavior of delay as a function of load and node density. First, we find the minimum delay (obtained under optimal transmission range) versus load for different number of nodes in the network. In FIGURE 5 we plot the results. FIGURE 6 shows average minimum delay versus N for different loads. As expected, average delay gets larger as the number of nodes increases. But, how fast does it increase? Can we predict or at least

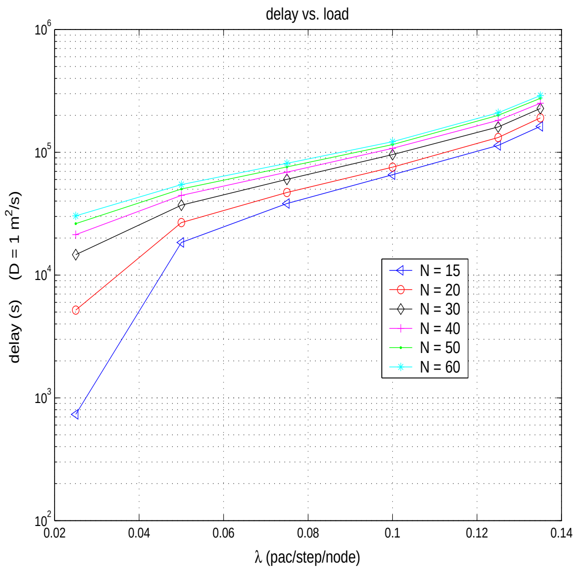

Fig. 5. Average packet delay as a function of load for different node densities. Diffusion coefficient, D, is taken as 1 and delay is plotted in seconds.

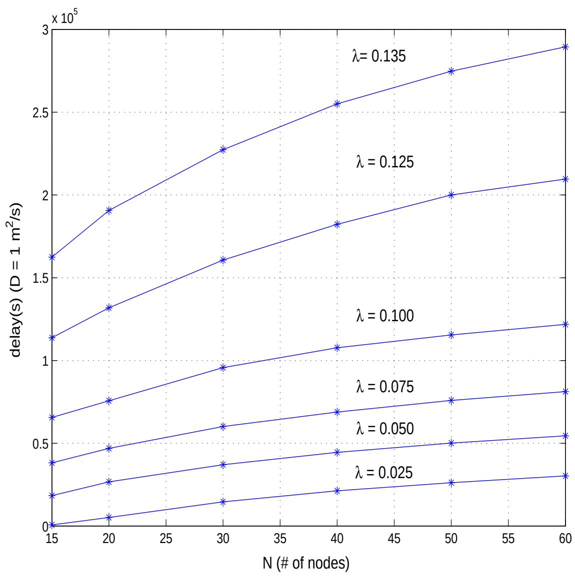

Fig. 6. Average packet delay as a function of number of nodes for different loads. Diffusion coefficient, D, is taken as 1 and delay is plotted in seconds.

bound the average delay for larger densities? We examine the high load case since the finite area effects are more severe under light load. FIGURE 7 shows average delay vs. N for packet generation rate $\lambda = 0.135$ in logarithmic scale.

Let us denote the data points of FIGURE 7 as (N_i, T_i) $i \in \{1, \ldots, 6\}$ and define the curve $y_i = c_i\sqrt{N}$ passing through (N_i, T_i). In FIGURE 7, we also plot all y_i curves and observe that for $\forall i \in \{1, \ldots, 6\}$:

$$T_j < y_i(N_i) \quad \text{if } N_j > N_i, \tag{21}$$

$$T_j \geq y_i(N_i) \quad \text{if } N_j \leq N_i \tag{22}$$

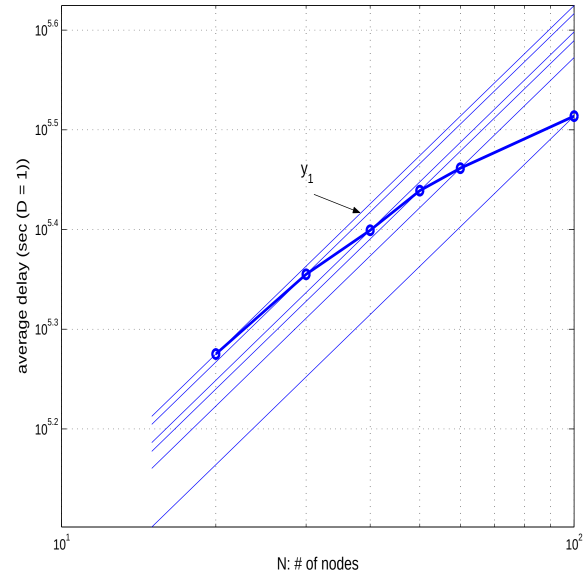

Fig. 7. Average packet delay as a function of N for $\lambda = 0.135$ (log scale).

Thus, we conclude that for all the data points obtained, average delay increases at a rate slower than $\sqrt{N}$. This result is in contrast with the delay of fixed networks. In fixed networks, under optimal routing, the number of hops between the nodes grows as $\sqrt{N}$ [1]. Since all the delay is due to transmission time, packet delay also increases as $\sqrt{N}$. In mobile infostations, although the delay is much larger than the generic ad hoc networks, its growth rate is smaller.

2) Scaling of the Throughput Capacity: It is difficult to determine the exact value of maximum throughput by simulation since it might require us to simulate the network for a very long time. However, we can invoke the arguments used in [2] and [3] to show that as N increases, the maximum throughput of the one-hop threshold strategy goes to a nonzero constant.

In a one-hop strategy, the throughput can be characterized by the probability of success of a node's transmission which is a function of the receiver SINR. We will show that the receiver SINR is not effected by rescaling of the network topology. Since rescaling is equivalent to changing node density, we can argue that the long-term network throughput of the system is independent of node density. The development below follows [3] except that [3] assumes constant transmit power while we choose different powers for different transmitter-receiver pairs.

Let us consider a network with node density, ρ_1, with area A_1 and shrink it in 2-dimensional space to obtain another network of area A_2 and density ρ_2. Both from [3] and simulation studies in [12], we know that the optimal transmission range also shrinks in the same ratio. Let us assume that a transmission takes place between the two nodes with probability 1 when their distance is smaller than the transmission range. When we make a connectivity graph of the two networks showing the transmitters and receivers, the two graphs will be scaled versions of each other [3], [13]. We denote the set of transmitters, same in both cases, as $\mathcal{T}$. If node-i is a transmitter, its receiver is denoted by node-i'.

Let us pick a random transmitter node and call it node-0. For both cases, we calculate the SINR at its receiver, node-$0'$, due to the transmission of node-0. Since the connectivity graphs are scaled version of each other, the SINR expressions are identical for the two networks:

$$\gamma = \frac{P_0/d_{oo'}^{\alpha}}{\sum\limits_{i\in\mathcal{T},i\neq 0} P_i/d_{i0'}^{\alpha}} \tag{23}$$

However, the distances and power levels are different in two cases. From equation (2), we recall that in our protocol a transmitter chooses its power level according to the distance to its receiver :

$$P_i = K d_{ii'}^{\alpha} \tag{24}$$

where $K = N_0 W \gamma^* / d_0^{\alpha}$. Substituting equation (24) in equation (23) we obtain:

$$\gamma = \left(\sum_{i\in\mathcal{T},i\neq 0} (d_{ii'}/d_{i0'})^{\alpha} \right)^{-1} \tag{25}$$

In equation (25), it is seen that the SINR does not depend on the exact distances but their ratios. As the two graphs are scaled versions of each other, ratio of these distances are the same in both networks. Thus, the SINR of the node does not change with scaling. This is due to the fact that the transmit power is also scaled with the distance and both signal and interference power are effected from scaling in the same way. So, we conclude that in one-hop threshold strategy the throughput capacity per node can be kept constant as the node density goes to infinity.

VII. Summary and Conclusion

This work was an initial attempt to understand the possible tradeoffs between mobility, throughput and delay. For computational and implementation purposes we avoided network-centric solutions and concentrated on a packet-oriented approach where the routing decisions are made for each packet independent of the other packets. We analyzed the average delay and throughput performance of our threshold-based policy. We resorted to network simulations to evaluate delay and throughput.

We have found the optimal cost parameters coefficients that minimize delay for a given network load. Results showed that most of the time our policy can be replaced with a one-hop threshold rule, which is equivalent to a threshold rule on the transmission range, without significant change in the average delay. Thus, allowing multiple hops has negligible benefit when the node moves in a somewhat unpredictable way.

Then we examined the one-hop threshold strategy and its scaling properties. We showed that the throughput per node can be kept scalable while the average delay increases at a rate smaller than $\sqrt{N}$, where N is the number of nodes.

References

[1] P. Gupta and P. R. Kumar. Capacity of Wireless Networks. *IEEE Transactions on Information Theory*, 46(2):388–404, March 2000.

[2] M. Grossglauser and D. Tse. Mobility Increases the Capacity of Ad Hoc Networks. *IEEE/ACM Transactions on Networking*, 10(4):447–486, Aug. 2002.

[3] W. H. Yuen and R. D. Yates. Optimum Transmit Range and Capacity of Mobile Infostation Networks. In *Proc. of Globecom*, 2003.

[4] F. Atay and C. Rose. Exploiting Mobility in Mobile Infostation Networks to Decrease Transmit Power. To appear in WCNC 2004.

[5] Eugene Perevalov and Rick Blum. Delay limited capacity of ad hoc networks: Asymptotically optimal transmission and relaying strategy.

[6] Nikhil Bansal and Zhen Liu. Capacity, delay and mobility in wireless ad-hoc networks. In *Infocom*, 2003.

[7] Renato M. de Moraes, Hamid Sadjadpour, and J.J. Garcia-Luna-Aceves. Increasing capacity and reducing delay through mobility in ad-hoc networks. Submitted to Infocom 2004.

[8] D. Bertsekas. *Dynamic Programming: Deterministic and Stochastic Models*. Prentice-Hall, 1987.

[9] R. G. Gallager. *Discrete Stochastic Processes*. Kluwer, 1996.

[10] L. Peterson and B. Davie. *Computer Networks*. Morgan-Kaufmann, 2000.

[11] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1993.

[12] F. Atay. Exploiting Mobility in Mobile Ad Hoc Networks. Master Thesis, Dept. of Electrical and Computer Engineering, Rutgers University, Piscataway, NJ, USA, 2003.

[13] R. Meester and R. Roy. *Continuum Percolation*. Cambridge University Press, 1996.