Kinetic Minimum-Power Routing and Clustering in Mobile Ad-Hoc Networks

Camillo Gentile Jerome Haerri Robert E. VanDyck
National Institute of Standards and Technology
Gaithersburg, Maryland 20899
{cgentile,haerri,vandyck}@antd.nist.gov

***Abstract*— We have previously developed a distributed routing algorithm which minimizes the number of overhead messages necessary to maintain the minimum-power multi-hop routes in a mobile ad-hoc network, assuming a piece-wise linear model for the motion of the nodes. In this paper we extend the routing algorithm to include clustering as well, to reduce further the number of overhead messages at the expense of sub-optimal routes. A single parameter controls the degree of clustering, and consequently the degree of sub-optimality, rather than arbitrary parameters such as maximum cluster size or maximum distance between nodes. The proposed algorithm converges in a £nite number of iterations to both stable routes and stable clusters, and by setting the cluster parameter to 0 collapses to the original routing algorithm with no clusters and optimum routes.**

I. Introduction

In a mobile ad-hoc network (MANET), many nodes are either battery operated or otherwise power-constrained. Therefore it is often necessary to maintain minimum-power routes between the nodes of the network as they move about, without a significant decrease in throughput attributed to overhead messages. By modeling node positions as a linear trajectory, as opposed to a £xed position for a single time instant, we have developed in previous work [1] a distributed algorithm that renders a priori (i.e. at the time of organization) the *kinetic spanning tree* [2], or the sequence of shortest-path trees between nodes for all time. The algorithm sustains virtually no increase in overhead messages compared to organizing the routes for a single time instant through the Bellman-Ford algorithm [3]. Moreover, the kinetic tree requires minimum updating only when a node changes course since most of the routes remain valid, rather than updating at uniform periods and re-propagating routes across the whole network.

While the kinetic routing algorithm minimizes the number of overhead messages, this number may still prove prohibitive as the size of the network grows. A popular approach to deal with issues of scalability suggests grouping nodes into clusters [4]-[6]: the ordinary nodes of a cluster elect a clusterhead and route through it to other nodes in the network, at the expense of sub-optimal routes. In this paper, we propose the extension of the kinetic routing algorithm to clustering as well. We study the tradeoff between minimum-power routing and clustering, and accordingly develop a kinetic algorithm which simultaneously organizes both the clusters and the routes of the network. The distributed algorithm minimizes a global multi-objective function, which allows control of the degree of clustering based on the minimum-power routing, promoting intelligent clustering by grouping into clusters nodes with similar trajectories, rather than arbitrary parameters such as maximum cluster size or maximum distance between nodes.

[0]This work was supported by the Advanced Research and Development Activity (ARDA) under contract number 706400.

II. Preliminaries

This section reviews the kinetic routing algorithm developed in our previous work. Assuming a piece-wise linear model for the motion of a node

$$\mathbf{x}_i(t) = \begin{bmatrix} x_i(t_0) + \dot{x}_i(t_0)\cdot t \\ y_i(t_0) + \dot{y}_i(t_0)\cdot t \end{bmatrix}, \quad (1)$$

where vector $[x_i(t_0),\ y_i(t_0)]^T$ denotes the position of node i at the initial time t_0, and vector $[\dot{x}_i(t_0),\ \dot{y}_i(t_0)]^T$ its velocity, the algorithm computes the kinetic spanning tree.

The squared distance between nodes i and j is

$$\begin{aligned} D_{ij}^2(t) &= \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|_2^2 \\ &= a_{ij}t^2 + b_{ij}t + c_{ij}, \end{aligned} \quad (2)$$

and the *power cost* as a function of time, required to transmit between nodes i and j, is de£ned as $P_{ij}(t) = P_{ji}(t) = \kappa D_{ij}^2(t)$, for some constant κ. The kinetic spanning tree translates to a forwarding table at each node whose entries are the minimum-power paths in time to the sink node. The table of node i appears in Fig. 1. The shaded area indicates its *composite power cost* $P_i(t) = \min_t\{P_{ij_1[]S}(t), P_{ij_2[]S,}(t), P_{ij_3[]S}(t)\}$ required to route from i to S through the next-hop nodes j_1, j_2, and j_3; [] denotes the ordered nodes comprising the rest of the path. Node i forwards to node j_1 for $0 \le t < t_1$, to node j_2 for $t_1 \le t < t_2$, to node j_3 for $t_2 \le t < t_3$, and to node j_2 for $t_3 \le t$.

As a simple example, Table I(a) lists the trajectories of four nodes routing directly to the sink node S at time $t_0 = 0$, except D $\rightarrow$ A. The changes in the kinetic tree, computed a priori at time $t_0 = 0$, appear in Table I(b).

III. Interaction of Routing and Clustering

A. Power Function

The kinetic routing algorithm described in the previous section effectively minimizes at node i the *power function*

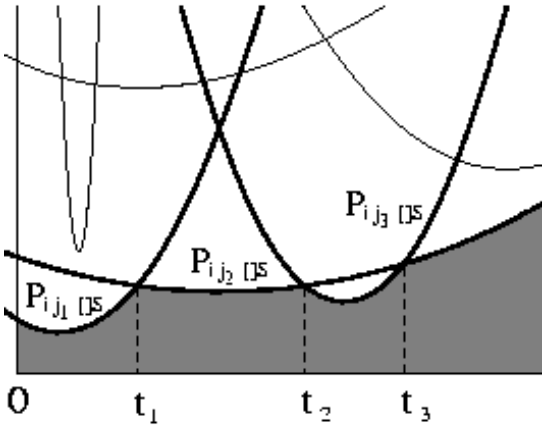

Fig. 1. New and current power costs in the pruning step at node j.

TABLE I

CHANGES IN THE KINETIC SPANNING TREE.

Node	Trajectory
S	$(0, 0)$
A	$(-1 - 0.1t, 1)$
B	$(-0.4t, 1.5 - 0.2t)$
C	$(2 - 0.5t, 1 + 0.2t)$
D	$(0, 2 + 0.3t)$

(a)

t	Change
1.252	A → B
2.193	D → C
3.754	A → S
4.085	B → A
5.692	C → A
8.378	D → A
8.452	D → S

(b)

$$F_i^P = \int_{t_0}^{\infty} P_i(t)\, dt \tag{3}$$

in a distributed manner, where t_0 denotes the execution time.

If clusterhead i joins another cluster j as an ordinary node, node i then routes to S through node j. Accordingly its composite cost $P_i(t)$ increases to $P_i'(t) = P_{ij}(t) + P_j(t)$ forcing an increase in F_i^P as well, hence sacri£cing the minimum-power routes to reduce the overhead messages necessary to reorganize the network in the event of a trajectory change. Note that if node j was routing through i before the clustering, that path no longer exists after the clustering, hence F_j^P (as well as the power function of other clusterheads routing through i) increases as well.

B. Reorganization Function

Given N nodes in a network and assuming on average m nodes per cluster, we determine the optimal value m^* which minimizes the cost of reorganizing the network in the event of a course change. Let n indicate the number of clusters, then $p_n = n/N$ represents the probability that a clusterhead changed trajectory conditioned on some node changing trajectory, with $R_n(m)$ the associated cost of reorganization. Likewise $p_m = (N - n)/N$ represents the conditional probability that an ordinary node changed trajectory, with associated cost $R_m(m)$. We formulate the conditonal cost of reorganization as

$$R(m) = p_n \cdot R_n(m) + p_m \cdot R_m(m). \tag{4}$$

The values of $R_n(m)$ and $R_m(m)$ depend on the chosen clustering protocol. Here we assume that ordinary nodes route directly (i.e. single-hop) to their clusterhead, while the clusterheads route through each other in a multi-hop fashion to the sink node. Also the sink node will always choose to stay a clusterhead to minimize the power function.

When an ordinary node changes course, the features used to group it with its cluster change, and so it may no longer belong to that cluster; accordingly, we reinitialize this node as a single clusterhead and allow it to reorganize. If a clusterhead changes course, we reinitialize it as a single clusterhead; in addition, the ordinary nodes associated with this cluster no longer have a route to the node, and so must reinitialize as single clusterheads and be allowed to reorganize.

The message complexity incurred to reorganize the multi-hop routes of n clusterheads to the sink varies as $\mathcal{O}(n^2)$ [7], hence for this particular protocol $R_n(m) = (n + m - 1)^2$ and $R_m(m) = (n+1)^2$. Substituting $n = N/m$ into (4) and solving yields the solution for m^* which minimizes $R(m)$

$$\begin{aligned} m^* &= \frac{N(N-4)}{\frac{1}{6}\sqrt[3]{2}\lambda + \frac{1}{3}\sqrt[3]{256}\frac{N^2}{\lambda} - \frac{2}{3}N}, \\ \lambda &= N^{\frac{2}{3}}[27N^2 - 248N + 432 + 3(N-4)\sqrt{3(27N^2 - 280N + 432)}]^{\frac{1}{3}}. \end{aligned} \tag{5}$$

The *reorganization function* below quanti£es the degree to which node i departs from the optimal con£guration: $R(m)$ is much steeper farther from m^* and so a smaller cluster will ben-e£t more from accepting an additional node than a larger cluster. In minimizing the function to 0, the network converges to m^* nodes per cluster.

$$F_i^R = \int_{\min\{m_i, m^*\}}^{\max\{m_i, m^*\}} R(m)\, dm \tag{6}$$

C. Multi-Objective Function

Before organization, each node in the network begins as a single clusterhead. If the network organizes through minimizing the power function alone, no clustering takes place and so constructs the kinetic minimum-power routes between the clusterheads to the sink. At the other extreme, minimizing only the reorganziation function results in a network with $n^* = N/m^*$ clusters to reduce the number of overhead messages to maintain the routes. In general, we seek a compromise between the two opposing functions by minimizing the multi-objective function

$$F_i = F_i^P + \delta \cdot F_i^R, \tag{7}$$

where the only parameter δ controls the degree of clustering suited to a particular application, rather than arbitrary parameters such as maximum cluster size or maximum distance between nodes [4]-[6]. F_i^R forces the merging of nodes, and subject to this constraint F_i^P gives preference to clustering those nodes moving with similar trajectories, since the power cost *in time* between them will be smallest. So the power function will cluster in an intelligent manner based on the feature of similar trajectories. We can imagine separate troops of soldiers moving in a military operation, all soldiers in a troop following similar

trajectories. Our proposed clustering based on minimum-power routing will favor making each troop a separate cluster, especially if there is a great distance between troops. Therefore the clustering can reveal the structure behind the network, if one exists.

As pointed out in Section III-A, when a node i joins another cluster, it alters not only the *local* objective function F_i, but also the function of other nodes in the network. Hence we minimize the global multi-objective function $F = \sum_i^N F_i$ through a distributed technique known as "region competition" [8]. The following section provides the details of the algorithm.

IV. Kinetic Routing and Clustering Algorithm

Initialization

Each node maintains a forwarding table for itself and its neighboring clusterheads (not ordinary nodes). It also maintains the number of nodes in each of their clusters. Each node j in the network begins as a clusterhead. It adds to its table j the power cost to the sink, $P_{jS}(t)$. If S is not a neighbor of j, then $P_{jS}(t) = \infty$. Node j distributes $P_{jS}(t)$ to its neighbors.

Iteration Step

1. UPDATE: At the current time step, node i receives $P_{j[]S}(t)$ from neighboring clusterheads $j \in j_1, \cdots, j_n$ and adds $P_{j[]S}$ to its table j.

2. PRUNE (clusterheads only): Node i computes the *current cost* $P_{ij[]S}(t) = P_{ij}(t) + P_{j[]S}(t)$ for each node j. The current costs and the previous costs (i.e. those in table i from the previous step) are compared amongst each other. Only those (minimum) costs which contribute to the composite cost $P_i(t)$ are retained in table i. Fig. 1 displays both the current and previous costs in table i at the current step. They appear as six parabolic functions in time, however only three of them, namely $P_{ij_1[]S}, P_{ij_2[]S}$, and $P_{ij_3[]S}$, contribute to $P_i(t)$. Only the current costs added to the table in the current step are then distributed to neighboring nodes, since the previous costs added to the table were already transmitted in those steps, and need not be retransmitted.

3. CLUSTER (single clusterheads and ordinary nodes only): At this step node i may join a neighboring cluster j, if such an action lowers the global multi-objective function of the network, which translates to a negative *change* over the local multi-objective functions of those nodes affected by the clustering. In region competition, node i computes for each of its n neighboring clusterheads $j \in j_1, j_2, \ldots, j_n$ the change in the global function $\Delta F[j]$ which results from joining cluster j. i then joins the cluster j^* which offers the most negative change. If there is no negative change amongst the candidate neighboring clusterheads, node i remains in its current state.

If the clustering takes place, i must be removed from the tables of all neighboring clusterheads which route through it to the sink, resulting in the modified $P'_{j_1}(t), P'_{j_2}(t), \ldots, P'_{j_n}(t)$ which can be used to compute the changes in the local functions of these affected nodes. Since i contains the routing table of all its neighboring clusterheads, i can compute these changes locally without any additional exchange of messages[1]. The change in the local function of tcalculation affected i is computed using the composite cost $P'_i(t) = P_{ij}(t) + P'_j(t)$ from routing directly through clusterhead j. In addition, one needs to consider the new $m'_i = m'_j = m_j + 1$ and $m'_{jk} = m'_j$ of the ordinary nodes of cluster j, as well as the corresponding changes in the number of nodes in the cluster that i leaves if it joins j[2]. If i joins j^*, it does so by sending one *join* message to its neighbors which achieves three objectives: 1. it informs j^* that it will be joining it; 2. it updates the value of m of the ordinary nodes in cluster j^* and the previous cluster of i; 3. it removes any entries on the tables its neighbors maintain that contain i on its path.

The reader should be aware of the special cases in which a *join* message is denied:

1. Node i attempts to join clusterhead j^*, which itself has joined another clusterhead as an ordinary node at the same iteration. In this case j^* sends a *deny* message and node i remains in its current state.
2. Two single clusterheads wish to join each other's cluster. In this case the node with the smaller ID (or some other convention may be used) joins the cluster of the node with the larger ID. No *deny* message is required.

Given the non-convexity of the global multi-objective function for $\delta > 0$, simulated annealing is introduced to avoid convergence to a local minimum. Note that with $\delta = 0$, the kinetic algorithm simply collapses to the kinetic routing algorithm.

V. Modified Power Cost

Employing the power cost described thus far clusters based on the state of the network as $t \to \infty$, since the last entry in the forwarding table of a node approaches ∞ as $t \to \infty$ (see $P_{ij_2[]S}(t)$ of Fig. 1). This attenuates the contribution of the other entries to $P_i(t)$ and to F_i^P. To avoid this problem, we bias the clustering towards t_0 since as $t \to \infty$ all the nodes will have changed trajectory anyway. We de£ne

$$p_i(t) = e^{-\beta_i(t-t_i)} \tag{8}$$

as the probability that a node i is continuing on its present trajectory, where the Poisson parameter $\frac{1}{\beta_i}$ indicates the average time the node follows a course, and t_i the time its current trajectory began.

Assuming independent node trajectories, $p_{ij}(t) = p_i(t) \cdot p_j(t)$ describes the probability that nodes i and j are continuing on their respective courses at time t. The modi£ed power cost below probabilistically weights $P_{ij}(t)$ to re¤ect the uncertainty of the link's existence.

$$\tilde{P}_{ij}(t) = -\frac{p_{ij}(t)}{P_{ij}(t)} \tag{9}$$

A low modi£ed power cost favors a low power cost with high probability. It readily substitutes in the algorithm described in

[1] An exact calculation would involve removing i from the tables of *all* clusterheads in the network, not just the neighboring, but this would not allow local computation of the change; moreover, the neighboring nodes are those most affected by the clustering.

[2] For a correct implementation of region competition, the two components of the change in the global function $\Delta F[j]$ must be normalized in the sum by the largest respective components of the candidates: $\frac{\Delta F^P[j]}{\max\limits_{j \in j_1, \cdots, j_n} \{|\Delta F^P[j]|\}}$ and $\frac{\Delta F^R[j]}{\max\limits_{j \in j_1, \cdots, j_n} \{|\Delta F^R[j]|\}}$.

the previous section by simply changing the alegbraic sum to the sequel:

$$\begin{aligned}\tilde{P}_{ij[]S}(t) &= \tilde{P}_{ij}(t) \oplus \tilde{P}_{j[]S}(t) \doteq -\frac{p_i(t) \cdot p_{j[]S}(t)}{P_{ij}(t) + P_{j[]S}(t)} \qquad (10)\\ &= -\frac{e^{-(\beta_i+\beta_{j[]S})(t-\frac{t_i\beta_i+t_{j[]S}\beta_{j[]S}}{\beta_i+\beta_{j[]S}})}}{(a_{ij}+a_{j[]S})t^2+(b_{ij}+b_{j[]S})t+(c_{ij}+c_{j[]S})}\\ &= -\frac{e^{-\beta_{ij[]S}(t-t_{ij[]S})}}{a_{ij[]S}t^2+b_{ij[]S}t+c_{ij[]S}} = -\frac{p_{ij[]S}(t)}{P_{ij[]S}(t)}\end{aligned}$$

Now the £ve parameters $a_{ij[]S}, b_{ij[]S}$, $c_{ij[]S}, \beta_{ij[]S}$, and $t_{ij[]S}$ describe $\tilde{P}_{ij[]S}(t)$ when sending a routing message, rather than three for $P_{ij[]S}(t)$.

It can be shown that that p_n and p_m in Section III-B are still valid under the assumption of Poisson-distributed trajectory changes, and so the associated equations presented to generate the reorganization function still prove valid.

VI. Simulation Results

The following section contains simulation results for the routing and clustering algorithm for a network of N nodes uniformly spread over a 10×10 mile area at initialization time $t = 0$. N varies as $\{20, 40, 80\}$ by simulation, and all results were averaged over 10 trials. We assign each node a random velocity, whose speed does not exceed 60 miles/hour. The transmission radius of the each node is limited to 4 miles. We assume that the underlying communication links requires 100 ms to transmit a packet across single link, and set the unit time for one iteration accordingly. The course changes of the nodes are Poisson distributed with value $\beta = 0.03$, hence a node changes its trajectory on average every 33s.

A. Fixed Trajectories

The experiments with £xed trajectories allow the reader to extract the intrinsic properties of the network characterized by the clustering parameter, namely the number of clusters formed, the number of routing messages required for self-organization, and the transmission power of the associated routes. Each simulation was run for 30 seconds, terminating before any expected trajectory changes.

An increase of the clustering parameter from the value 0 forces the reorganization function to play a more signi£cant role in the multi-objective function, emphasizing the cost of routing messages to organize the network. This attracts clusterheads to join other clusters as ordinary nodes, reducing the total number of clusterheads and so the $\mathcal{O}(n^2)$ complexity of the algorithm accordingly. Fig. 2 illustrates the structure of the organized network for the values $\delta \in \{0.0, 0.2, 0.5, 0.7, 1.0\}$. The optimal number of clusterheads for $N \in \{20, 40, 80\}$ given through (5) is $n^* \in \{1.72, 2.23, 2.90\}$ respectively, and we see in each case that the number of clusters approaches n^* as δ increases.

For the same operation points of the clustering parameter, Fig. 3 displays the corresponding number of routing messages required for self-organization of the network from initialization to convergence to stable routes. The savings in the number of messages with increasing δ is more pronounced as the size of the network grows, from a factor of 23% with $N = 20$ to a factor of 448% with $N = 80$. It is worth mentioning that the iteration complexity of the algorithm, hence the number of iterations required for convergence, varies in proportion to the message complexity. For $\delta = 0$ and $N \in \{20, 40, 80\}$ the algorithm requires on average $\{5, 10, 20\}$ iterations for convergence, each decreasing with larger values of the clustering parameter.

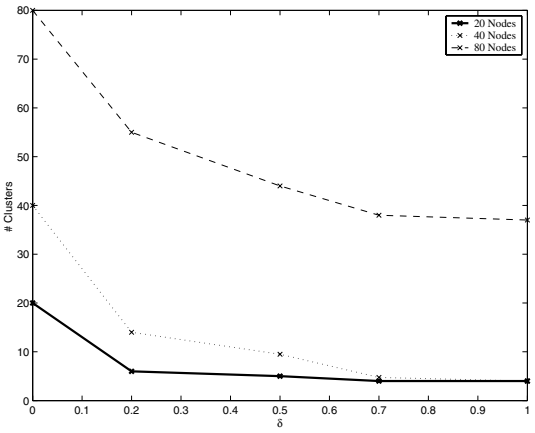

Fig. 2. Number of clusters formed vs. the clustering parameter with £xed trajectories.

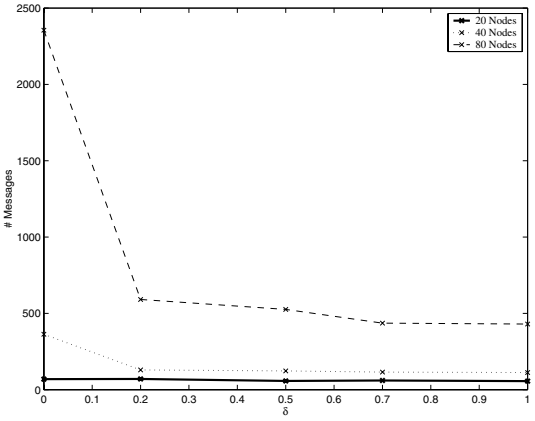

Fig. 3. Number of routing messages vs. the clustering parameter with £xed trajectories.

Fig. 4 displays the sum of the power function over the N nodes in the network versus δ, normalized by this quantity for $\delta = 0$; hence the transmission power associated with the minimum routes assumes unity. The plot illustrates the departure from the minimum-power routing as δ increases from 0. As expected, the power increase for routing as δ increases becomes more pronounced for larger values of N. For $\delta = 1$, the increase is only 55% for $N = 20$, but grows to 545% for $N = 80$.

These results show the tradeoff between minimum-power and the number of routing messages, suggesting the operation of the network at an application-suited point characterized by the value of δ.

B. Randomly Changing Trajectories

The simulations presented in this subsection are identical to those in the previous subsection, except that here we run them

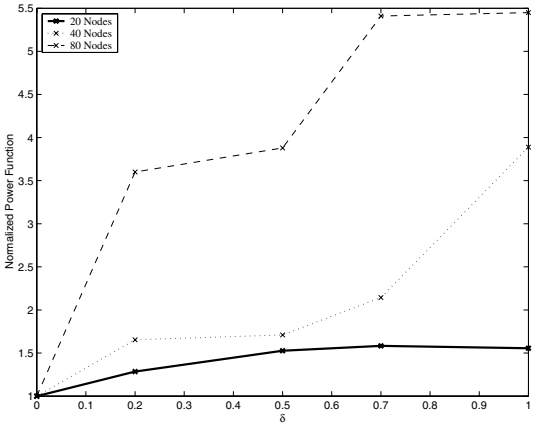

Fig. 4. Normalized power function vs. the clustering parameter with £xed trajectories.

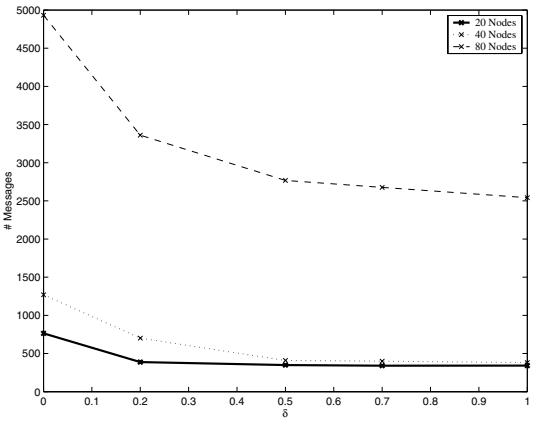

Fig. 5. Number of routing messages vs. the clustering parameter with changing trajectories.

for an extended duration of £ve minutes, and moreover the nodes are allowed to change trajectory according to the Poisson distribution with parameter $\beta = 0.03$, and reorganize accordingly. We present the same format of results in the number of messages required to organize and maintain the routes throughout the simulation period, and the associated normalized power function over all the nodes in the network. The normalized power function is summed from the intialization time $t = 0$ to $t = 300s$. We cannot report the number of clusters formed throughout the run since this value ¤uctuates with every course change.

Fig. 5 displays an even more dramatic reduction in the number of routing messages required in the presence of trajectory changes as δ increases. Unfortunately such a reduction results in a corresponding drastic increase in the power requirements for transmitting from all nodes in the network to the sink with sub-optimal routes, as re¤ected in Fig. 6.

VII. Conclusions and Current Work

We have described a distributed algorithm which simultaneously organizes the routing and clustering of a network, exploiting the theory of kinetic spanning trees. As in kinetic routing algorithm introduced previously, the proposed algorithm converges in a £nite number of iterations to both stable clusters and

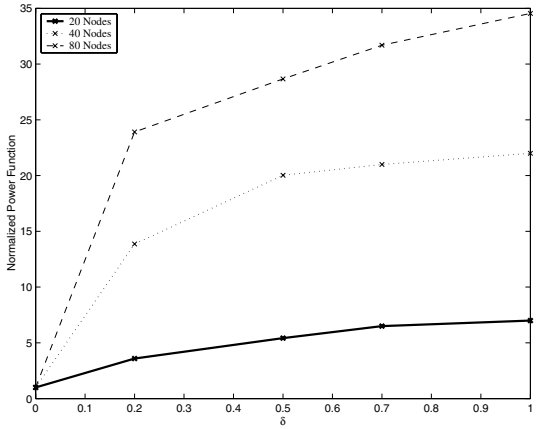

Fig. 6. Normalized power function vs. the clustering parameter with changing trajectories.

stable multi-hop routes between the clusterheads. Moreover, the organization updates are event-driven by a change in the trajectory of a node, as opposed to updating at uniform periods as in conventional algorithms.

The simulation results show that by increasing the clustering factor δ, we can signi£cantly decrease the number of messages necessary to organize and maintain the routes from all the nodes in the network to the sink node. This reduction, however, comes at the expense of sub-optimal routes in terms of power tranmission.

Our current work involves simulating networks where the nodes move about as groups according to similar trajectories, rather than independently according to some random distribution. The simulations should illustrate the true utility of organizing the network by minimizing the multi-objective function, where the groups are formed intelligently into functional clusters.

References

[1] C. Gentile and R.E. Van Dyck, "Kinetic spanning trees for minimum-power routing in MANETS," *Proc. Vehicular Tech. Conf. Spring*, May 2002, pp. 1415-1419.

[2] P. K Agarwal, D. Eppstein, L. J. Guibas, and M. R. Henzinger, "Parametric and kinetic minimum spanning trees," *Proc. 39th Annual Symp. on Foundations of Computer Science*, pp. 596-605

[3] R. G. Gallager, P. A. Humblet, and P. M. Spira, "A distributed algorithm for minimum-weight spanning trees," *ACM Trans. on Prog. Lang. and Systems*, Vol. 5, No. 1, pp. 66-77, Jan. 1983.

[4] L. Subramanian and R. Katz, "An architecture for building self-con£gurable systems," *IEEE Conf. on Mobile and Ad Hoc Networking and Computing*, Aug. 2000, pp. 63-73.

[5] J.-H. Ryu, S. Song, and D.-H. Cho, "New clustering schemes for energy conservation in two-tiered mobile ad-hoc networks," *IEEE Conf. on Communications*, Jun. 2001, pp. 862-866.

[6] B. An and S. Papavassiliou, "Mobility-based clustering for mobile ad-hoc wireless networks," *Conf. on Information Sciences and Systems*, Mar. 2001, pp. 60-64.

[7] K.V.S. Ramarao and S. Venkatesan, "On £nding and updating shortest paths distributively," *Journal of Algortihms*, Vol. 13, pp. 235-257.

[8] S.C. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation," *IEEE Trans. on PAMI*, Vol. 18, No. 9, pp. 994-900, Sept. 1996.