Event Triggered Distributed Collaborative Control

John S. Baras* and Pedram Hovareshti*
Department of Electrical and Computer Engineering
and Institute for Systems Research
University of Maryland, College Park
College Park, MD 20742, USA
{baras, hovaresp}@umd.edu

Senni Perumal
AIMS, Inc.
6213 Executive Blvd.
Rockville, MD 20852, USA
senni.perumal@aims-sys.com

Abstract— **We consider the collaborative control of a group of autonomous mobile agents. Building upon our earlier work we consider the communication needs and connectivity of the agents' network as they move. We develop algorithms that automatically sense the possibility of connectivity loss among the agents. We also consider the automatic detection of path disconnection when more than one path need to be maintained between pairs of agents. Using local probing schemes we formulate such problems as event-triggered control problems. We develop distributed algorithms that automatically select some agents and move them appropriately so as to maintain certain degree of desired connectivity among the moving agents. We characterize the trade-off between the gain from maintaining a certain degree of connectivity vs. the combined cost of communications and the associated dynamic re-positioning of agents. The results illustrate the efficiency achieved by event-triggered control in such problems. We also describe the resulting communication topologies and in particular their similarity to dynamic small world topologies.**

I. INTRODUCTION

Control of swarms of autonomous platforms (vehicles, robots, aircraft, ships, etc.) has recently attracted great interest due to a wide variety of applications such as providing coverage and connectivity to ground agents, automated highway systems, mobile sensor networks, disaster relief efforts, collaborative robotics, etc.

In problems regarding the control and coordination of vehicle networks, decentralized methods are preferred as centralized control requires immense communication and computational resources The collaborative control of autonomous mobile agents can thus be viewed as a hierarchical design problem: A high level decision making and path planning module is responsible for maintaining the connectivity while creating a sequence of way-points for their motion. A low level motion control module computes the real control commands for agents to follow the way-points generated under the motion constraints. In a series of previous papers (see e.g. [19] [20]) Baras, Tan and Xi addressed the problems of high level and low level motion planning using a combination of distributed hybrid Gibbs sampler based methods and model predictive control (MPC).

In this paper, we address the problem of designing a high level component responsible for maintaining the communication needs of the group, and in particular the (path-) connectivity of their communication network as they move. The module has outputs to, and inputs from the higher level path planning component . This communication connectivity is crucial in scenarios with a group of agents moving in a particular area and covering it, while avoiding obstacles and collisions. Here the connectivity is maintained by clustering the agents and providing connections between the clusters. Building on our previous work [14], we address the problem of maintaining connectivity among ground clusters of moving agents. Among the agents in each cluster, one is designated as the *cluster-head*. The cluster heads can be designated in a distributed manner [2], and can be equipped with multi-mode communication capabilities. In normal situations, connectivity is maintained by the cluster-heads, who send messages directed to other cluster-heads. Therefore, the connectivity of clusters is a crucial factor in the performance of the group of agents as a whole. As a result, we develop algorithms that sense the possibility of loss of connectivity among the agents. When direct communication between the clusters is not possible, a suggested solution is to use Aerial Platforms (APs) as relays in networks. However, the use of APs is costly and should be kept to a minimum level. Furthermore, the APs should be positioned so that the resulting network is well-connected. In other applications specific agents can move to specific locations and/or change their physical characteristics (e.g. increase their transmission power) in order to provide and maintain communication connectivity between the moving agents.

The organization of the paper is as follows. Section II provides the basic set up and discusses a model for detecting the possibility of loss of connectivity among the agents. Section III formulates the problem of maintaining connectivity as an event-triggered control problem, which uses centralized decision making. In section IV we explore the issues of collaborative decision making at the onset of connectivity failure between clusters. Section V discusses a clustering algorithm used to provide the intended connectivity between the ground units using as few APs as possible. Section VI investigates the question of characterizing efficient and 'good' topologies, which provide better connectivity.

*Research supported by the National Aeronautics and Space Administration (NASA) Marshall Space Flight Center under cooperative agreement no. NCC8-235, and also by the U.S. Army Research Laboratory Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011.

II. Basic set up

We consider a number of autonomous mobile agents in a terrain. Using the methods from our previous work [19], the planning algorithm initially moves the nodes, so that they are arranged in n clusters. We assume that each group has a leader (cluster-head), who is in charge of maintaining communications with the other leaders. From now on we refer to the leaders as nodes or agents. The nodes have identical omnidirectional antennas with good quality communication possible if their distance is less than a constant R_{disc}.

The communication connectivity of the agents is modelled by an undirected graph. The vertices $v_i \in V,\ i = 1, 2, ..n$ denote the agents and $e_{ij} = (v_i, v_j) \in E,\ i, j = 1, ..., n$ denote the links between them. Assume that there are overall m links between the agents (leaders); we give an arbitrary ordering to the links and alternatively represent the link set as $E = \{l_k, k = 1, ...m\}$. We denote the set of nodes who are at graph distance k from node v_i as its $k-$neighbors (i.e. $k-$hops away from node v_i, in multi-hop paths without loops). We also call the set of nodes with graph distance less than or equal to k from node v_i, and the edges between them as node i's $k-$ neighborhood and denote it by $\mathcal{N}_k(v_i)$. A graph is called $k-$connected if the minimum number of edge removals needed to make the graph disconnected is k.

We assume that initially the leaders form a (path) connected graph, and by sending out HELLO messages, each leader knows its neighbors and has an estimate of their distance [21]. Also each leader knows its two-hop away neighbors. The nodes may occasionally become disconnected. This may happen because of sudden obstacle occurrences or deviations from previously determined paths upon terrain uncertainties. We assume that the agents send distress messages (help request signals) to the AP or some other coordinating unit, if they think AP intervention is necessary to save the graph connectivity. However, AP intervention is costly and should be considered only if the link losses affect the connectivity in a serious manner. The basic problem that each agent has to address is whether to call for intervention or not, but before that, each agent has to be provided with a method to sense the possibility of link losses.

The agents periodically monitor their own energy level with respect to a reference threshold to make sure if they can transmit correctly. If they sense any decline in energy or other resources, they will try to inform the coordinating unit (or the AP) directly or via their neighbors. Each agent sends periodical HELLO messages to their neighbor. Upon receiving a HELLO message a node sends an ACK message to the sender. If a node does not receive an acknowledgement after a timeout interval, it will resend the HELLO message. We set a maximum threshold N_{try}. If the number of trials exceeds this threshold, the link is considered lost [21]. Recent literature has addressed elaborate distributed fault detection in wireless systems and other distributed systems [1] and [4]. Nodes can employ various local monitoring mechanisms to monitor connectivity or loss of connectivity: SNR measurements, SINR measurements, packet transmission success rates, levels of received power and their variations, etc. Here, we assume that, as in [4], the nodes are provided with failure detection modules by which they can maintain reachable node lists that enable them to detect unwanted partitioning and network disconnection. We use this simple event-triggered model, and study the tradeoff between asking for AP intervention *vs.* the cost of risking graph disconnection.

III. A model for AP intervention cost analysis

In this section we consider the problem of deciding whether AP intervention is necessary in the presence of link losses. The problem is formulated as a stopping time problem in a stochastic control framework. We assume that local probing schemes enable the nodes to keep track of the reachable nodes and of losses in connectivity. We also assume that the link/path disconnection is reported to a coordinating unit (or to the AP). The coordinating unit decides on whether AP intervention is necessary or not.

A. General model

Consider a $k-$ connected network configuration. Assume a time horizon N and that during each time interval t, each link can fail with some probability. The failure probabilities are determined by terrain specifications and the nodes' power levels. For simplicity we assume a constant average disconnection probability p for all links. The 'state' of the system at each time t is denoted by x_t and is equal to the number of link deletions necessary to make the graph disconnected.

There are two possible choices for the control action at each time interval. The choice of $u_t = 0$ corresponds to normal system operation, in which no AP intervention is demanded. In this case the state transitions as a Markov system based on the transition probabilities resulting from the graph topology and link deletion probabilities. The choice of $u_t = 1$, corresponds to AP intervention, in which case APs will be sent to circumvent the lack of connectivity problem. We augment the state space with a termination state T. After the AP intervention, the state will transition to T and will remain there for the rest of the time horizon.

In the normal system operation, we consider a connectivity cost C_{CON}, which is a non-increasing function of k, the edge-connectivity of the graph. We assume that the engagement and operation of the APs causes a constant rate cost C_{AP}. We model this by a stopping cost. At the time $t^\star$, when the stopping action happens ($u_{t^\star} = 1$), the system incurs a cost of $C_{AP}(N - t^\star + 1)$. The system evolves as:

$$x_{t+1} = \begin{cases} w_t & \text{if } u_t = 0 \\ T & \text{if } u_t = 1 \end{cases} \tag{1}$$

where $w(t)$ is determined by the graph topology and link loss probabilities as described above. The problem could be posed as an optimal stopping time problem, via defining a cost function:

$$g(x_t, u_t, w_t) = \begin{cases} C_{AP}(N - t) & \text{if } u_t = 1 \\ C_{CON}(w_t) & \text{if } u_t = 0. \end{cases}$$

The dynamic programming (DP) equations are:

$$J_t(x_t) = \begin{cases} \min[C_{AP}(N-t+1), E[C_{CON}(w_t) \\ \qquad\qquad\qquad\qquad +J_{t+1}(w_t)], & \text{if } x_t \neq T \\ \qquad\qquad\qquad\qquad\qquad 0 & \text{if } x_t = T. \end{cases}$$

The coordinating unit (or the AP(s) themselves) decides on whether to send APs based on this stopping criterion.

B. *The ring case*

We now illustrate the approach by considering a group of n moving agents with a ring communication topology. The agents participate in a mission with time horizon N. The agents periodically send messages and based on their received messages, they calculate estimates of distances and update their list of reachable nodes. We assume independent link losses. Since a ring is a $2-$ connected topology, the state space S consists of three actual states and a termination state T, $S = \{0, 1, 2, T\}$.

If we denote the probability of state transition from i to j using control $l, (l = 0, 1)$ before the stopping as $p_{ij}(l)$, then because of the independence assumption, for $i = 0, 1, 2$, $p_{ij}(l)$ can be easily calculated, e.g. $p_{i0}(0) = 1 - (1-p)^n - np(1-p)^{n-1}$.

In the normal mode of operation, when the AP is not called in, we consider the cost to be proportional to the average hop length. We assume that when AP intervenes, there is a constant cost of C_{AP} per operation cycle. If the agents fail to call for intervention before a disconnection happens, there would be a high cost $C_{disc} > C_{AP}$. If the intervention happens, we assume that the APs guide the agents to their nominal trajectories. The incurred costs are:

$$g(x_t, u_t) = \begin{cases} C_{AP}(N-t) & \text{if } u_t = 1 \\ 2n & \text{if } u_t = 0, x_t = 2 \\ 4n & \text{if } u_t = 0, x_t = 1 \\ C_{disc} & u_t = 0, x_t = 0 \end{cases}$$

Therefore the problem is cast as a stopping time problem and it can be solved using the DP iteration:

$$J_t(x_t) = \min\{C_{AP}(N-t+1), E[g(x_t) + J_{t+1}(w_t)]\}.$$

with the terminal cost $J_N(x_N) = C_{AP}$. The optimal cost to go function $J_k(x_k)$ is a monotone decreasing function of its argument. We have simulated the ring scenario with different numbers of nodes, maintenance and disconnection cost, and error probabilities. Figure 1 illustrates the benefit of the event triggered AP calling for periodic surveillance. The parameters of the simulation are $p = 0.01, n = 10, N = 20, C_{disc} = 300$. Nodes start with complete ring topology and are subject to losses. We have allowed the termination cost to vary, $C_{AP} = \alpha C_{disc}$, for $\alpha \leq 1/3$. We have compared the expected cost of the event triggered scheme, with the cost of periodic AP presence, in which an AP is considered to always be present and subject to a cost of C_{AP} per time interval. Figure 1 shows the ratio of cost improvement due to using the event triggered method ($\frac{J_0(2)}{N.C_{AP}}$),with respect to $\alpha = \frac{C_{AP}}{C_{disc}}$ for different number of nodes. For smaller n, there is a cut-off at small values of α demonstrating that even when the cost of surveillance is small, it is beneficial to use the event triggered method. However, for a fixed probability of link loss p, as the number of nodes in the ring increases, the probability of failure increases. For larger n, with fixed p, the cut-off happens at greater values of α.

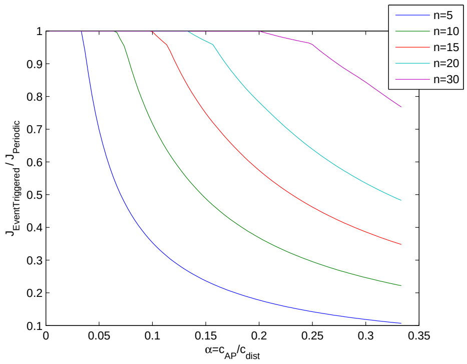

Fig. 1. Event triggered cost improvement ratio $\frac{J_0(2)}{N.C_{AP}}$ vs. $\alpha = \frac{C_{AP}}{C_{disc}}$

IV. A Collaborative method for maintaining connectivity

Here we propose a rating mechanism by which, each agent rates its incident links (edges) and the links in their $k-$neighborhood. The “importance” rate that a node v_i assigns to an incident edge l_j will be an indicator of the number of paths starting from v_i, which pass through l_k. After each link is rated by its incident links, a consensus type algorithm can be utilized, so that a single average “importance rate” is assigned to each link. If a set of links fail, our algorithm decides on how many APs are needed to intervene based on the “importance” of the failing links.

A. *Measures of link importance*

The importance of a link is a measure of how many disjoint paths in the network use it. The following notions of “between-ness” [11] capture the importance of links for our application:

- *Shortest path between-ness*: The number of shortest paths between all pairs of nodes in a network which pass through the suggested link.

However, calculation of the above between-ness measure is a centralized task. Here we provide a decentralized, local information based algorithm, with which the nodes can rate their incident links. This way the links will be associated with different importance rates from nodes in their neighborhood. If necessary a consensus-type algorithm will be used to attain a common estimate of the links’ importance. If a set of links fail, the decision on whether and how many APs to send, will be taken using the information on the importance of links which have failed. We will now describe the algorithm.

B. *A local scheme for finding the importance of the links*

We assume that each node always maintains a list of its current neighbors. We also assume that each node keeps a list of nodes which are reachable through its $k-$neighbors. The choice of k is a trade-off between the tractability and locality of the scheme. Therefore we prefer the use of small values of k (e.g. k= 1,2,3). This can be done by neighborhood discovery methods (e.g. [4], [21]). The idea is that each node periodically sends heartbeat signals [1] to its neighbors and piggybacks (on these heartbeat signals) the set of paths of which it is aware of since sending its last heartbeat signal. Upon receiving a message, a node investigates the paths through each of its neighbors and adds the nodes in each path, to the set of reachable nodes through its neighbors.

For a known number k, each node v_i forwards messages along all of the paths in its $k-$neighborhood. The message will be propagated throughout the network, so that all possible nodes will receive the message. When a node receives a message, it updates its reachable set along the corresponding path in its $k-$ neighborhood. If this procedure continues, the nodes will eventually come up with a list of reachable set of nodes through each path in their $k-$neighborhood. Each node then rates each of the paths. The rating r_{ik} of each path p_k, according to node v_i is equal to the number of nodes which are reachable from v_i through p_k. The nodes then rate each edge in their $k-$neighborhood, where E_{ij}, the rating that each node associates to link j, is equal to the sum of the ratings of all paths emanating from v_i, which include link l_j; i.e. $E_{ij} = \sum_{l_j \in p_k} r_{ik}$.

After each node runs the algorithm, it will assign subjective importance ratings, to all of the links in its $k-$neighborhood. Therefore each node v_i will maintain a list E of the importance values $E_{ij}(k)$ it has assigned to the links l_j in its $k-$neighborhoods. A normalized list E_{norm} is a measure of the relative importance of the links in the $K-$neighborhood of each node. The main properties of the algorithm that follow from the definition can be summarized as:

Propostion 1: The following statements hold:

1) An edge l will be rated by a node v if and only if $l \in \mathcal{N}_k(v)$, i.e. l falls in the $k-$ neighborhood of v.
2) The importance rating decreases monotonically along a path emanating from a rater node.

C. Link loss report based on link importance

Since our scheme provides nodes with subjective importance ratings on links in their $k-$neighborhood, these ratings should be used in each node's decision making on asking for AP intervention. When a link fails, we assume that the nodes incident to it, broadcast the link loss to their $k-$neighbors. Upon acknowledging a link loss in a node's $k-$neighborhood, the node can decide on whether to ask for intervention or not based on its importance rating (or normalized importance rating) assessment of the lost link.

V. Clustering

Assume that all the ground leaders have the same altitude (of 0) and form M clusters ($C_j, j = 1, \ldots, M$). Aerial Platforms (APs) placed appropriately and acting as relays can be used to provide connectivity between the M disconnected ground clusters. Since APs are scarce/expensive resources, the goal is to find the minimum number of APs and their locations so that the resultant network (both between the nodes and the APs and between the APs) is connected.

The ground nodes and the APs have identical omni-directional radios with the signal between nodes decaying as $1/R^\alpha$ where R is the distance between nodes and α is the path loss exponent, which depends on the environment between the nodes. The radio specifications and the path loss exponent α together determine a maximum communication distance between the nodes. α is equal to 2 (i.e., free space communication) for communication between the ground agents and APs as well as for communication between the APs. This results in a maximum communication distance of R_2 between the ground agents and APs and among the APs. Since ground nodes communicate with α strictly greater than 2 ($\alpha = 4$ for a suburban environment), the maximum communication distance R_0 between leaders is strictly less than R_2 (usually by an order of magnitude). Assume that all the APs fly at an altitude of h such that the maximum communication distance between agents and APs projected onto the ground, R_1 (given by $R_1 = \sqrt{R_2^2 - h^2}$), is greater than R_0. Thus the problem of finding the minimum number of APs (L) and their positions can be reduced to $\mathcal{R}^2$, with a_j denoting the position of the APs projected onto the ground.

A. Problem Formulation

We formulate the connectivity problem as a constrained clustering problem ([15], [16]) with a summation form distortion function ($D(C, A)$) involving the distances between the ground clusters (C) and the APs (A) and a summation form cost function ($C_1(A)$) involving only the distances between the APs (A). The resultant clustering problem is then solved using Deterministic Annealing (DA) to obtain near-optimal solutions. In order for the ground nodes and the APs to form a connected network, we need: 1) At least one node from each cluster within a radius of R_1 from an AP; and 2) Each AP is within R_2 of some other AP (i.e., the APs form a connected graph).

Assuming that the APs are numbered from 1 to L, we can make sure that they form a connected network by ensuring that any AP numbered j is connected to at least one lower numbered AP i, where $i < j$. This is used in the DA solution where when we add a new AP, we make sure that it is connected to at least one of the previously added APs. Hence the connectivity problem can be stated as:

$$
\begin{gathered}
Minimize\ L; \quad subject\ \ to \\
\exists a_1, \ldots, a_L; \quad \max_{j \in \{1,\ldots,M\}} \quad \min_{\substack{g \in C_j \\ i \in \{1,\ldots,L\}}} \quad \| \ g - a_i \ \| \leq \ R_1 \\
\text{and,} \qquad \max_{l \in 2,\ldots,L} \quad \min_{m<l} \quad \| \ a_l - a_m \ \| \leq \ R_2
\end{gathered}
$$

where $\| \ g - a \ \|$ is the l^2-norm between points g and a on the ground. Finding the exact solution to the problem above involves an exhaustive search on the different ways in which nodes can be selected from each cluster and the ways clusters can be grouped together for coverage by

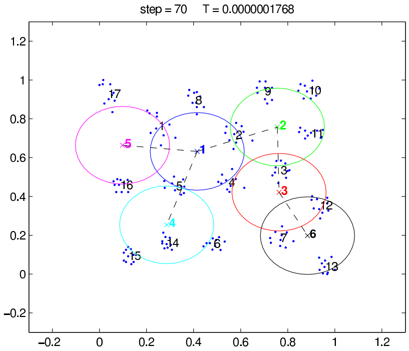

Fig. 2. Complex Scenario: AP Placement with AP-ground node connectivity and AP-AP connectivity.

a single AP all the while making sure that the APs are connected to each other. This problem is NP-hard as it is a generalization of the Euclidean *disk-cover* problem. Hence using the approximation,

$$\max(s_1, \ldots, s_n) \cong (s_1^\alpha + \ldots + s_n^\alpha)^{\frac{1}{\alpha}} \quad \text{for large } \alpha$$

we can convert the AP-ground node and AP-AP constraints into a summation form,

$$\begin{aligned} & Minimize\ L; \quad subject\ to \\ \exists a_1, \ldots, a_L; \quad & \sum_{j=1}^{M} d_1(C_j, a_{u_1(j)}) \quad \leq \quad R_1^\alpha \\ \text{and,} \quad & \sum_{l=2}^{L} d_2(a_l, a_{u_2(l)}) \quad \leq \quad R_2^\beta \end{aligned}$$

for large α and β, where,

$$\begin{aligned} d_1(C_j, a_i) &= \min_{g \in C_j} \parallel g - a_i \parallel^\alpha \\ d_2(a_l, a_m) &= \min_{m<l} \parallel a_l - a_m \parallel^\beta \end{aligned}$$

and $u_1(j)$ is a function that assigns an AP to every cluster; $u_2(l)$ is a function that assigns the closest lower numbered AP to an AP.

Constrained clustering problems of the above form are non-convex optimization problems except in special cases. Hence the Deterministic Annealing (DA) method is used to solve the constrained clustering problem for globally near-optimal solutions. Within the framework of constrained clustering ([15], [16]), the distortion function between the ground nodes and the APs is given by $D(C, A) = \sum_{j=1}^{M} d_1(C_j, a_{u_1(j)})$ and the cost function among the APs is given by $C_1(A) = \sum_{l=2}^{L} d_2(a_l, a_{u_2(l)})$. Figure 2 illustrates an application of our algorithm.

VI. Topology

By adding APs we have the advantage that the previously far apart nodes now communicate through APs. Prior to AP addition the neighborhood relation was based on physical proximity. The addition of APs extends the concept of neighborhood in that two far apart agents can communicate directly through their corresponding APs. We assume that each AP is capable of short time high energy transmission upon necessity and that via a suitable medium access control and AP energy scheduling, agents which are located geographically far from each other can communicate.

This extended notion of neighborhood makes long range edges realizable. However, establishing long range connections requires higher cost. Therefore there is a trade off between the cost of maintaining long range connectivity and group performance. Here we consider two classes of 'efficient' topologies, small world graphs and expander graphs.

A. *Small world graphs*

The small world graphs based on the model of Watts and Strogatz [18] take a regular lattice and replace some original edges by random ones connecting nodes at 'long distance' with some probability $0 \leq \phi \leq 1$; i.e. by introducing 'short cuts'. This family of graphs shows a favorable trade off between performance and cost of collaboration.

In [8] Higham analyzed the small world phenomenon in the Watts-Strogatz model by considering the hitting time of a slightly randomly perturbed Markov chain on a ring. Building on [8], we studied consensus problems on grid-based small world graphs in [2]. We showed a significant speed up in the convergence speed of consensus algorithms in Watts-Strogatz models compared to that of the grid base by perturbing the consensus weight matrix. The perturbation corresponds to considering rare transitions among non-neighboring states in the Markov chain associated with the grid. In [9], we showed that by choosing shortcuts with low probability one can improve the convergence rate of regular grids significantly in a probabilistic sense. For other interesting works on small world graphs in the control community and consensus applications see e.g. [17], [12], [13], [6].

Here, we consider a general setting in which the base graph can be any sparse graph. We are interested in the question: is a given graph "small-worldizable"? The following procedure gives an implicit definition of what we mean by being "small-worldizable". Other criteria for a graph to be small-worldizable are investigated by [5].

Recall that if F is a primitive stochastic matrix, according to the Perron-Frobenius theorem [3], $\lambda_1 = 1$ is a simple eigenvalue with a right eigenvector $\mathbf{1}$ and a left eigenvector π such that $\mathbf{1}^T \pi = 1$, $F^\infty = \mathbf{1}\pi^T$ and if $\lambda_2, \lambda_3, ..., \lambda_r$ are the other eigenvalues of F ordered in a way such that $\lambda_1 = 1 > \mu = |\lambda_2| \geq |\lambda_3| \geq ... \geq |\lambda_r|$, and m_2 is the algebraic multiplicity of λ_2, then

$$F^t = F^\infty + O(t^{m_2-1}|\lambda_2|^t) = \mathbf{1}\pi^T + O(t^{m_2-1}|\lambda_2|^t)$$

Then $\Delta = 1 - \mu(F)$ denotes the spectral gap and linear iterations on graphs with higher spectral gaps converge faster.

Definition 1: Small-worldizable graphs Given a connected graph $\mathcal{G}_n$ on n vertices:

- Consider a natural random walk on this graph. Denote the corresponding Markov Chain graph as

$$F_0 = (I + D)^{-1}(A + I)$$

where A is the adjacency matrix of the graph $\mathcal{G}_n$ and D is the diagonal matrix with each node's degree on the corresponding diagonal.

- Perturb the zero elements of F_0 by $\epsilon < \frac{1}{n}$ and adjust it to get a new stochastic matrix F_ϵ.

$$(F_\epsilon)_{ij} = \begin{cases} \epsilon & (F_0)_{ij} = 0 \\ (1-n\epsilon)(F_0)_{ij} + \epsilon & (F_0)_{ij} \neq 0 \end{cases}$$

i.e. $F_\epsilon = (1-n\epsilon)F_0 + \epsilon\mathbf{1}\mathbf{1}^T$.

- $G_n(V,E)$ is small-worldizable if $\frac{\Delta(F_\epsilon)}{\Delta(F_0)} \gg 1$, where $\Delta(F)$ denotes the spectral gap $1-\mu(F)$.

Small-worldizability can be characterized by:

Theorem 1: A graph $\mathcal{G}$ is small-worldizable if and only if $\frac{\mu}{1-\mu} \gg \frac{1}{n\epsilon}$.

The proof of Theorem 1 is a direct result of the following Lemma:

Lemma 1: The second largest eigenvalue modulus (SLEM) of F_ϵ is given by

$$\mu(F_\epsilon) = (1-n\epsilon)\mu(F_0)$$

.

Proof: (Sketch) Consider the matrix

$$F_1 = (1-n\epsilon)^{-1}F_\epsilon = F_0 + \frac{\epsilon}{1-n\epsilon}\mathbf{1}\mathbf{1}^T.$$

From the Sherman-Morrison-Woodbury formula we have

$$\det(F_1 - \lambda I) = [1 + \frac{\epsilon}{1-n\epsilon}\mathbf{1}^T(F_0-\lambda I)^{-1}\mathbf{1}]\det(F_0-\lambda I) \tag{2}$$

Furthermore, for any $\lambda \notin Spec(F_0)$,

$$(F_0-\lambda I)^{-1}\mathbf{1} = (1-\lambda)^{-1}\mathbf{1}$$

It follows that the eigenvalues of F_1 are the same as the eigenvalues of F_0 except for $\lambda_1(F_1) = 1 + \frac{n\epsilon}{1-\epsilon}$. Therefore: $\lambda_1(F_\epsilon) = 1$, and for $i \neq 1$, $\lambda_i(F_\epsilon) = (1-n\epsilon)\lambda_i(F_0)$. The result follows.

■

As an example consider the case of ring-type base graphs of n nodes where each node is connected to its k neighbors [18]. By exploiting the circulant structure of ring type graphs $C(n,k)$, for $k \ll n$, we can derive the result that $\Delta(F_0) = O(n^{-2})$. By considering $\epsilon = n^{-\alpha}, \alpha > 1$, it can be shown that the small world effect holds for $\alpha = 2$. At $\alpha = 3$, the effect of shortcuts begins to dominate. This is the onset of the small world phenomenon.

B. Expander graphs

Expander graphs have certain properties that make them suitable for our application: their large spectral gap ensures fast routing and convergence of decentralized algorithms; the path diversity they provide results in robustness to link failures [10]. There are at least two methods for distributed construction of AP-level expander graphs:

- Following the approach of [10] we can form a random expander graph as a $2d-$regular multi-graph in which the set of edges consists of d separate Hamiltonian cycles on the APs. Such a graph can be constructed distributedly and its diameter will be $O(\log_d n)$ with high probability.
- Following the approach of [7] we can form a random expander graph as the union of two spanning trees chosen independently from the uniform distribution over all spanning trees. This can be implemented simply by taking a random walk and include edges that visit previously unvisited nodes. Such a graph has a constant edge expansion with high probability.

References

[1] M. K. Aguilera, W. Chen and S. Toueg,"On Quiescent Reliable Communication", *SIAM Journal on Computing,* 29(6): 2040–2073, 2000.

[2] J.S. Baras and P. Hovareshti, "Effects of Topology in Networked Systems: Stochastic Models and Small worlds", *Proc. of the 47th IEEE Conference on Decision and Control*, 2973–2978, 2008.

[3] P. Bremaud, *Markov chains, Gibbs Fields, monte Carlo Simulations and Queues.* Springer Verlag, 2nd edition, 1999.

[4] D. Conan, P. Sens, L. Arantes, and M. Bouillaguet, "Failure, Disconnection and Partition Detection in Mobile Environment", *Proc. of the Seventh IEEE International Symposium on Network Computing and Applications*, 119–127, 2008.

[5] P. Duchon, N. Hanusse ,E. Lebhar, and N. Schabanel, " Could any graph be turned into a small-world?", *Theoretical Computer Science*, 355(1): 96-103, 2006.

[6] F. Fagnani and S. Zampieri, "Randomized consensus algorithms over large scale networks", *IEEE Journal on Selected Areas in Communication*, 26(4) 634–649, 2008.

[7] N. Goyal, L. Rademacher, S. Vempala, "Expanders via Random Spanning Trees", *Proc. of the Nineteenth ACM-SIAM Symposium on Discrete Algorithms*, 576–585, 2009.

[8] D.J. Higham, "A matrix perturbation view of the small world phenomenon", *Siam Journal on Matrix Analysis and Applications*, 25(2): 429–444, 2003.

[9] P. Hovareshti, J. S. Baras, and Vijay Gupta, "Average Consensus over Small World Networks: A Probabilistic Framework", *Proc. of the 47th IEEE Conference on Decision and Control,* 375–380, 2008.

[10] C. Law and K. Siu, "Distributed construction of random expander networks" *Proc. of IEEE Infocom Conference*, 2133–2143, 2003.

[11] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks", *Phys. Rev. E* , 69:02611, 2004.

[12] R. Olfati Saber, "Ultrafast consensus in small-world networks", *Proc. of the American Control Conference*, 4:2371–2378, 2005.

[13] S. Patterson, B. Bamieh, and A. El Ebadi, "Distributed average consensus with stochastic communication failures", *Proc. of 46th IEEE Conference on Decision and Control*, 4251–4220, 2007.

[14] S. Perumal, J. S. Baras, C. J. Graff and D. G. Yee, "Aerial Platform Placement Algorithms to Satisfy Connectivity, Capacity and Survivability Constraints in Wireless Ad-hoc Networks", *Proc. of Military Communications Conferece (MILCOM08),* 1–7, 2008.

[15] K. Rose and E. Gurewitz and G. C. Fox, "Constrained Clustering as an Optimization Method", *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(8): 785–794, 1993.

[16] K. Rose and Miller, "Constrained clustering for data assignment problems with examples of module placement", *Proc. IEEE Int. Symp., Circuits and Systems,* 1937–1940, 1992.

[17] A. Tahbaz-Salehi and A. Jadbabaie, "Small world phenomenon, rapidly changing markov chains, and average consensus algorithm", *Proc. of 46th IEEE Conference on Decision and Control,* 276–281, 2007.

[18] D.J. Watts and S.H. Strogatz, "Collective dynamics of small-world networks", *Nature*, 393:440–442, 1998.

[19] W. Xi, X. Tan and J.S. Baras, "Gibbs Sampler-Based Self-Organization of Autonomous Swarms", *Automatica*, 1107-1119, July 2006.

[20] W. Xi and J. S. Baras, "MPC Based Motion Control of Car-like Vehicle Swarms", *Proceedings of 15th IEEE Mediterranean Conference on Control and Automation,* 1–6, 2007.

[21] J.S. Baras, V. Tabatabaee, P. Purkayastha, and K. Somasundaram, "Component Based Performance Modelling of Wireless Routing Protocols", submitted for publication, also *ISR Technical Report TR 2008-27, University of Maryland,* 2008.