Self-Adaptive Ad-Hoc/Sensor Network Routing with Attractor-Selection

Kenji Leibnitz*, Naoki Wakamiya*, Masayuki Murata*
*Osaka University, Graduate School of Information Science and Technology
1-5 Yamadaoka, Suita, Osaka, 565-0871 Japan
Email: {leibnitz|wakamiya|murata}@ist.osaka-u.ac.jp

***Abstract*— In this paper we propose MARAS, a biologically-inspired method for routing in a mobile ad-hoc/sensor network environment. We assume that all nodes have no explicit knowledge of the network topology, except for their coordinates and the neighboring nodes within an RF transmission range. MARAS then selects the next hop for forwarding a packet towards a destination node which is best suited depending on some measured metric values. The benefit of our proposed method is its ability to operate entirely in a self-adaptive manner and that it can easily compensate for sudden changes in the topology of the network.**

I. Introduction

Routing in mobile ad-hoc networks has recently attracted many researchers due to the flexibility in controlling the topology. Unlike conventional Internet routing which is constrained by the underlying OSPF routing functionality, ad-hoc networks and sensor networks use wireless links and can be configured in a highly dynamic way. While we consider in this paper ad-hoc and sensor networks in the same fundamental way, their actual requirements differ significantly. Ad-hoc nodes may be mobile computers or PDAs that are simply connected in an ad-hoc manner, whereas, sensor networks usually have limited computational capabilities and a limited lifetime due to exhausted battery power. Our focus in this paper lies on generic ad-hoc network architectures.

In general, there are two major reactive routing concepts for ad-hoc networks: *source routing*, e.g. in Dynamic Source Routing (DSR) [1], and *distance vector*-based, e.g. Ad-Hoc On-Demand Distance Vector (AODV) [2]. When a new route to a destination node is requested, the source routing approach uses probe packets to determine the path from source to destination node and stores this information in each packet. On the other hand, in AODV each node uses routing tables to maintain the information of forwarding nodes. The routing tables are also set up by flooding the network with probe packets. Several variants of DSR and AODV have been proposed to consider multiple paths between source and destination to increase transmission reliability, cf. [3].

In this paper we consider a geographical routing scheme for ad-hoc networks. This means that we assume that each node is capable of determining its own location as well as an estimate of the destination node location. Although our method is also capable to operate without this restriction, it clearly helps to reduce the number of packets flooded in the network by having a rough idea in which direction the packet should travel. The selection of the next hop is performed with the biologically inspired *adaptive response by attractor selection* (ARAS) concept [4].

The remainder of this paper is organized as follows. In Section II we briefly introduce related work on routing in ad-hoc networks, especially with focus on randomized methods. This is followed in Section III by a description of the proposed biologically-inspired approach and we elaborate on how to apply it for self-adaptively determining the next hop in routing. The proposal is evaluated in Section IV by simulation results and future extensions are discussed in Section V.

II. Related Work

While many proposals for ad-hoc routing have been published, we are mainly interested in a robust and flexible concept that can self-adaptively react to sudden changes in the network topology. A possible approach to enforce resilience is to deviate from deterministic routing algorithms and to perform the selection of the next hop in a probabilistic manner. Since the selection is done independently by each node, these ad-hoc routing schemes scale well with the number of nodes.

In order to simplify the task we assume that the hop forwarding mechanism uses geographical information, i.e., the nodes are aware of their own locations. The advantage of using location information in routing is that the number of packets required for searching the paths to the destination node can be limited in the direction of the destination node, cf. LAR (Location-Aided Routing) [5]. However, location based routing may have problems if the selection of the next hop is performed in an entirely greedy way. Packets must be routed around void areas where no forwarding node exists in the direction to the destination. For instance, GPSR (Greedy Perimeter Stateless Routing) [6] switches from greedy forwarding to routing around the perimeter of the void region when one is encountered. Finding the destinations can also be performed using specific location servers which are queried by nodes to find the destination [7].

Another class of random forwarding protocols is represented by GeRaF (Geographic Random Forwarding) [8] and ExOR [9]. Here, each node also assumes some limited geographic information of itself and the sink node. When a node wishes to forward a packet toward the destination, it broadcasts it to its neighboring nodes, which then cooperate among each other to determine the best choice as next hop.

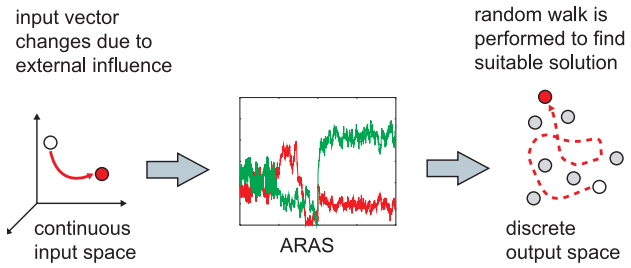


Fig. 1. General concept of ARAS

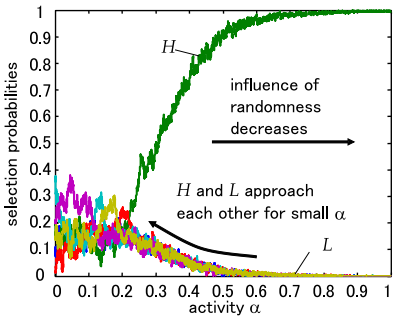


Fig. 2. Influence of α on output probabilities

III. The Proposed MARAS Mechanism

We propose MARAS (*mobile ad-hoc routing with attractor selection*), an entirely distributed multi-path routing mechanism, where each node operates autonomously and only gathers information about the network topology by exchanging messages with its neighbors. MARAS uses the concept of *adaptive response by attractor selection* (ARAS) for each node to select its next hop. In contrast to this paper, previous work in the field of overlay networks [10], [11] uses ARAS to self-adaptively select the primary path among a set of multiple predetermined paths at the source node. Those paths are obtained by using an underlying routing protocol on network layer and ARAS is used there to reduce the selfishness of flows in order to improve the overall performance of the network.s

A. Adaptive Response by Attractor-Selection

Adaptive response by attractor selection is a biologically inspired method for adaptively selecting one of several candidates which best reflects the current situation in a dynamic environment. ARAS is originally a model for its host E. coli cells to adapt to changes in the availability of a nutrient for which no molecular machinery is available for signal transduction from the environment to the DNA [4].

Basically, we can outline the attractor selection method as follows. Using a set of differential equations, we describe the dynamics of an M-dimensional system. Each differential equation has a stochastic influence from an inherent Gaussian noise term. Additionally, we introduce an *activity* $\alpha \in [0, 1]$ which changes the influences from the noise terms. For example, if α is large, the system behaves rather deterministic and converges to attractor states defined by the structure of the differential equations. However, for small α the noise term dominates the behavior of the system and essentially a random walk is performed. When the input values (*nutrients*) require the system to react to the modified environment conditions, activity α changes accordingly causing the system to search for a more suitable state, see Fig. 1. This may involve that α causes a previously stable attractor to become unstable.

Consider a set of A_n nodes of which we wish to select one to forward the packet as next hop of a node n. Let the cardinality of set A_n be M. For each node $i \in A_n$ we define the proportion of selecting node i as m_i with m_{max} being the maximum over all m_i. The dynamic behavior of m_i is characterized by the stochastic differential equation system given in Eqn. (1) for $i = 1, \ldots, M$.

$$\frac{dm_i}{dt} = \frac{syn(\alpha)}{1 + m_{max}^2 - m_i^2} - deg(\alpha)\, m_i + \eta_i \qquad (1)$$

The functions $syn(\alpha)$ and $deg(\alpha)$ are the rate coefficients of mRNA synthesis and degradation in the original biological model, respectively. They are both functions of α, which represents *cell activity* or vigor. The terms η_i are independent white Gaussian noise inherent in gene expression.

$$syn(\alpha) = \alpha\,[\beta\,\alpha^{\gamma} + \varphi^*] \qquad deg(\alpha) = \alpha \qquad (2)$$

The parameters β and γ in (2) are factors which influence the mapping of activity to the output probabilities and we use $\beta = 50$ and $\gamma = 3$ throughout this study. The variable φ^* is a special offset point which we will discuss below. For the sake of simplicity we also define $\varphi(\alpha) = \frac{syn(\alpha)}{deg(\alpha)}$.

When we define the functions $syn(\alpha)$ and $deg(\alpha)$ as given in (2), we obtain M equilibrium solutions $\mathbf{x}^{(k)}$ of Eqn. (1) in the form of

$$\mathbf{x}^{(k)} = \left[x_1^{(k)}, \ldots, x_M^{(k)}\right]^T \qquad k = 1, \ldots, M$$

with components $x_i^{(k)}$, see (3).

$$x_i^{(k)} = \begin{cases} \varphi(\alpha) & i = k \quad (H \text{ value}) \\ \frac{1}{2}\left[\sqrt{4 + \varphi(\alpha)^2} - \varphi(\alpha)\right] & i \neq k \quad (L \text{ values}) \end{cases} \qquad (3)$$

The behavior can be summarized as follows. The system in Eqn. (1) converges to solutions which have a single "high" value (H) and all other values are "low" (L). The dynamics of activity α influences the selected values. When α is high, the high value H also approaches 1.0, i.e., the selection becomes more deterministic. On the other hand, for small α, H and L become equal and the probabilities for selecting the next hop is controlled by the noise term, see Fig. 2.

Note that at $\varphi^* = 1/\sqrt{2}$ we have a special point, as the solutions $\mathbf{x}^{(k)}$ are only defined when $\varphi(\alpha) \geq \varphi^*$ (i.e., $H \geq L$). For $\varphi(\alpha) = \varphi^*$ we obtain a single solution $\mathbf{x}$ with the same entries. Finally, the resulting state vector is normalized to yield the probabilities for selecting a neighboring node as the next hop.

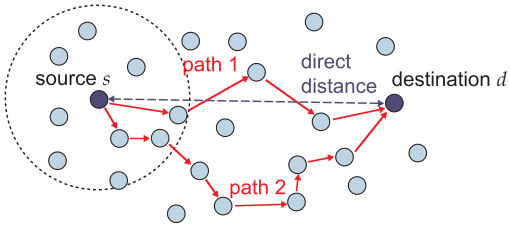


Fig. 3. Comparison of activity evaluation for two paths

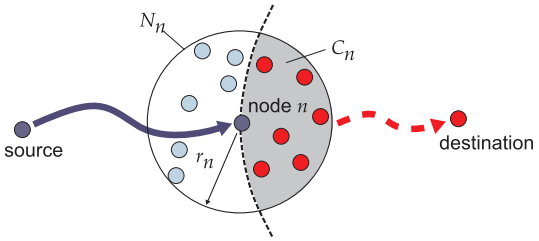


Fig. 4. Decision of next hop with ARAS

B. Mapping of Activity

The activity value α reflects the "goodness" of the current solution of ARAS. It is evaluated at the destination node and fed back to all nodes along the taken path. Its desired behavior is summarized as follows. If we have no information about which hop to choose (e.g. when selecting the hops for an initially unknown destination) the selection should be performed uniformly among all neighbors. Therefore, each node should initialize $\alpha = 0$ as this corresponds to the "no preference" case. The same applies to the case when the maximum number of hops (*maxhops*) is exceeded. A low α means that the current solution is not suitable and a new one should be searched. If a path to the destination exists, we wish to keep it with an activity of $0.0 < \alpha < 1.0$. The larger α is, the greater is the difference between H and L, which corresponds to the case where we have a clearly identified optimal choice as path as shown in Fig. 2. Therefore, we use in the following experiments an α as defined in (5).

$$\alpha^* = 1 - \left(1 - \frac{dist(s,d)}{len}\right)\left(1 - \frac{h_{min}}{h}\right) \tag{4}$$

$$\frac{d\alpha}{dt} = \delta\left(\alpha^* - \alpha\right) \tag{5}$$

The first term in (4) is the ratio of the direct distance $dist$ between source s and destination d over the path length len found by ARAS. Thus, α^* is small when the difference between both paths is large (i.e. the current path deviates much from the shortest, direct connection). In the second term, $\frac{h_{min}}{h}$ is the fraction of the previously found minimal number of hops over the current hops. Thus, by using this activity mapping, we achieve that short paths with a small number of hops are preferred over long paths. For example, in Fig. 3, path 1 is preferred over path 2, which leads to a higher activity feedback for path 1. Finally, δ is the rate of adaptation of α which we keep constant at $\delta = 0.1$. Finally, in order to avoid outdated information, activity decays over time.

C. Selection of the Candidate Nodes

Consider a scenario as shown in Fig. 4 where a source node transmits packets to the destination node of which it only knows its coordinate location. Each intermediate node decides autonomously with its own ARAS mechanism to which of its neighbors it forwards the packet. Once a packet has been delivered successfully (or not), the quality of the path is evaluated and used to update the decision of the node to maintain the current path or to dismiss it.

Let us assume the set of all nodes as $\mathcal{N}$ and an arbitrary node $n \in \mathcal{N}$ which receives a packet for the destination node $d \in \mathcal{N}$. On reception of the packet, the node identifies its current environment, by sending a distance request to its neighboring nodes. In order to reduce traffic load, this request can be done at a certain time interval. However, the more frequently this query is performed, the more accurate is the information about the current network topology. All neighboring nodes i reply to this request by reporting their respective distances $dist(i,d)$ to the destination. Based on this information, the node n maintains its neighbors in two logical sets: the *neighbor set* N_n and the *candidate set* C_n. The neighbor set contains all nodes within the transmission radius r_n of node n, i.e.,

$$N_n = \{i \in \mathcal{N} | dist(i,n) < r_n\}. \tag{6}$$

On the other hand, the candidate set is a subset of the neighbor set of all nodes which are nearer to the destination than n, i.e.,

$$C_n = \{i \in N_n | dist(i,d) < dist(n,d)\}. \tag{7}$$

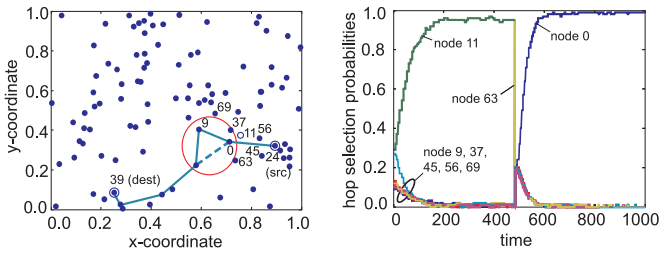
All nodes in C_n are potential candidates for forwarding, since the packet would get closer towards the destination with each hop. Which one is chosen is up to ARAS itself and will be described in the following section. For simplicity. we also define the complementary candidate set w.r.t. its neighbor set as $\overline{C}_n = N_n \backslash C_n$.

D. Extension of the Basic Algorithm

We can now summarize the basic algorithm for packet forwarding with MARAS when a node n receives a packet for destination node d:

1) If $n = d$, calculate α^* from (5), update all nodes along the path, and process packet at destination d.
2) Determine neighbor set N_n and candidate set C_n.
3) If $C_n = \emptyset$, then there is no suitable candidate in the direction of the destination. In order to avoid getting stuck in dead ends, we then set the ARAS set $A_n = \overline{C}_n$. Otherwise, i.e., if $C_n \neq \emptyset$, set $A_n = C_n$.
4) Perform ARAS on set A_n and forward packet to next hop according to resulting probabilities.

The dynamic behavior of the algorithm is illustrated in Fig. 5. There are $|\mathcal{N}| = 100$ nodes randomly distributed in a unit area size of $[0,1] \times [0,1]$. Each node has a transmission radius of $r = 0.3$. A snapshot of the system is given in Fig. 5(a). The source node (24) continuously sends packets to



(a) Network layout (b) Next hop probabilities of node 24

Fig. 5. Example of simulation scenario

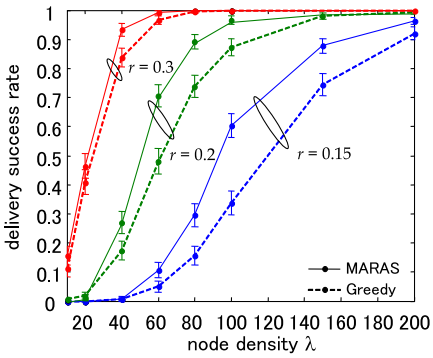


Fig. 6. Delivery rate over node density

the destination node using the previously described mechanism to determine its next hop. In Fig. 5(b) the next hop selection probabilities of the source node over the first 1000 time steps are given. Until about time step 500, node 11 is first chosen as relay hop. Then, this node leaves the system and the next hop is immediately switched to node 0 as shown in Fig. 5(a). The lower number of nodes in the candidate set leads to a higher selection probability of the primary hop candidate. Furthermore, although the probability for selection is about 0.95, the next hop is still chosen randomly, leading to slight variations in the path for each packet being transmitted. This also assists in resolving sometimes not optimal paths as highlighted in Fig. 5(a).

In order to actively remove such zigzag-shaped paths, we propose the following step when the activity of each node along the path is updated (step 1). Since this update is performed by a packet sent from the destination back to the source in the reverse direction of the original path, it is possible to check in the stored path in the packet if the i-th node along the original path has the j-th node ($i + 2 \leq j$) within its transmission range. All nodes of the path lying between these two nodes are eliminated simply by removing them from the respective sets A_n. The result of this operation would be the dashed line shown in Fig. 5(a).

IV. Numerical Evaluation

In the following we will evaluate the influence of the parameter settings on the performance of our method. The nodes are randomly distributed according to a spatial homogeneous Poisson process with density λ in a unit square, see [12]. We construct the process with increasing x coordinate and choose as source the node with the smallest x coordinate and as destination the one with the largest. Since the randomness of the node locations has a great influence, a large number of simulation replications is required. Thus, the following results are averaged over all connections of 500 simulation runs with 3000 time steps each. Error bars in the curves indicate the 95% confidence levels of the sampled mean values.

We compare the results of our method to those obtained from a simple greedy selection of the next hop. In the greedy method, each node forwards the packet deterministically to its neighboring node which lies nearest to the destination. It is expected that when the node density is large enough, greedy will always outperform our proposal since the latter will use a random decision for the next hop. Our focus therefore lies on the more interesting cases when the node density is low or the radius of the nodes is small.

A. Rate of Successful Delivery

Figure 6 shows the average packet delivery rate as a function of the node density λ. When the node density is very small, there is often no connectivity to the destination which leads to a high failure rate regardless of the routing method. We can see in Fig. 6 that MARAS achieves better performance than the greedy approach especially when the radius is small.

B. Resilience to Topology Changes

In order to cope with sudden changes in the network topology, we now also take the state of the nodes into account. Since we are interested how well the system performs in the presence of suddenly changing topologies, we let all nodes in the transit area with an x coordinate between 0.25 and 0.75 become inactive with a probability q. When q is large, many transit nodes will be unavailable for routing so the mechanism must adapt to find a new route to the destination.

It can be recognized in Fig. 7 that the delivery rate with MARAS is higher than that of the greedy approach due to the greater flexibility in the selection of the next hop. This effect is even greater for smaller node radius r.

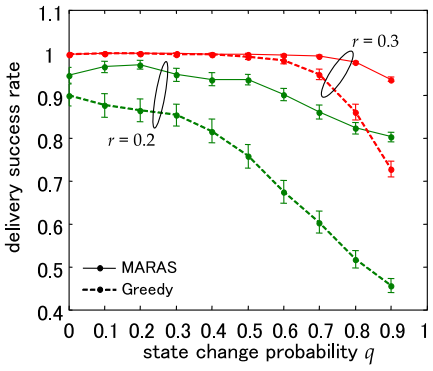


Fig. 7. Delivery rate over state-change probability

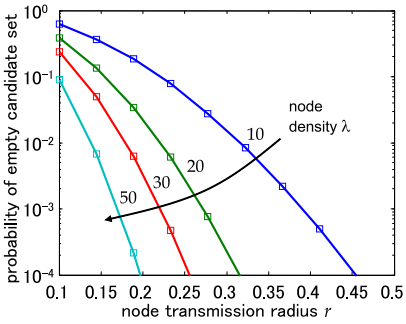


Fig. 8. Probability of finding an empty candidate set

C. Analytical Discussion on Density vs. Radius

The radius and density greatly influence the probability of finding a next hop within the coverage range. Due to the assumption that all nodes are distributed as a homogeneous Poisson process with density λ, we can analytically elaborate further on this. Assume that a node n is randomly located in the plane at distance d from the destination node. Our algorithm tries to find a next hop node in its coverage range r that lies closer to the destination than itself. Using simple geometry, the intersection area of the circle around n with radius r and the circle around the destination node with radius d can be given as $V(r,d)$

$$V(r,d) = \arccos\left(\frac{\tilde{X}}{r}\right) r^2 + \arccos\left(\frac{d-\tilde{X}}{d}\right) d^2 - d\,\tilde{Y}$$

with $\tilde{X} = \frac{r^2}{2d}$ and $\tilde{Y} = \sqrt{r^2 - \tilde{X}^2}$. The number of nodes in area $V(r,d)$ follows a Poisson distribution with rate λ, so the probability to find an empty candidate set of nodes is given as in Eqn. (8) and illustrated in Fig. 8 for $d = 1.0$.

$$P(K=0) = e^{-\lambda\pi V(r,d)} \tag{8}$$

In Fig. 8 we can recognize that for densities of $\lambda < 50$ and small radius r, there is a high probability of finding no node in the candidate set. In these cases the complementary candidate set $\overline{C}_x$ is used for MARAS, which may result in long detours of paths or dead ends.

V. Conclusion and Outlook

In this paper we presented a new approach for self-adaptive routing in an ad-hoc network. The concept is inspired from biology and is capable of rapidly reacting to changes in the environment. Basically, the proposed mechanism determines the probabilities for choosing the next hop of a packet on its path to the destination. Suitable paths with small number of hops or high path length-to-distance ratio are rewarded, whereas long paths or those which do not lead to the destination are penalized. The whole mechanism is controlled by an activity term which evaluates the current path and is fed back from the destination to all nodes along the path. Although we focused here on finding short paths, our method can easily take into account further metrics, e.g. radio link quality (signal-to-interference ratio), load of each node, etc. Numerical results indicate that the proposed method operates well when the node density and transmission radius are sufficiently large. Removal of nodes as well as node additions can be easily compensated.

The implementation itself can be performed in a straightforward manner with numerical methods which makes it also applicable for networks with nodes that have only limited computational capabilities like sensor networks. We can easily consider the total energy consumption per path and the residual energy of each node for selecting the path with the least energy dissipation. More detailed studies on appropriate input metrics for evaluating the paths as well as performance comparisons to other existing randomized mechanisms are the subject of future studies. Especially, when we consider a high churn rate of the nodes or high mobility, we expect that our method operates well compared to other approaches. Additionally, due to the randomization of the hop selection, a better load distribution will be achieved,

Acknowledgments

This research was supported by “The 21st Century COE Program: *New Information Technologies for Building a Networked Symbiosis Environment*”, “Special Coordination Funds for Promoting Science and Technology: *Yuragi Project*”, and a Grant-in-Aid for Scientific Research (A)(2) 16200003 of the Ministry of Education, Culture, Sports, Science and Technology in Japan.

References

[1] D. B. Johnson, “Routing in ad hoc networks of mobile hosts,” in *Proc. of the Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA, December 1994, pp. 158–163.

[2] C. Perkins and E. Royer, “Ad hoc on-demand distance vector routing,” in *2nd IEEE Workshop on Mobile Computing System and Applications*, New Orleans, LA, Feb. 1999.

[3] S. Mueller, R. P. Tsang, and D. Ghosal, “Multipath routing in mobile ad hoc networks: Issues and challenges,” *Lecture Notes in Computer Science*, vol. 2965, pp. 209–234, Jan. 2004.

[4] A. Kashiwagi, I. Urabe, K. Kaneko, and T. Yomo, “Adaptive response of a gene network to environmental changes by attractor selection,” *submitted for publication*, 2006.

[5] Y. B. Ko and N. H. Vaidya, “Location-aided routing (LAR) in mobile ad hoc networks,” in *Proc. Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, October 1998, pp. 66–75.

[6] B. Karp and H. T. Kung, “Greedy perimeter stateless routing for wireless networks,” in *Proc. of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000)*, Boston, MA, August 2000, pp. 243–254.

[7] J. Li, J. Jannotti, D. S. J. De Couto, D. R. Karger, and R. Morris, “A scalable location service for geographic ad hoc routing,” in *Proc. of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000)*, Boston, MA, August 2000.

[8] M. Zorzi and R. Rao, “Geographic random forwarding (GeRaF) for ad hoc and sensor networks: multihop performance,” *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, pp. 337 – 348, 2003.

[9] S. Biswas and R. Morris, “ExOR: Opportunistic multi-hop routing for wireless networks,” in *Proc. of ACM SIGCOMM*, Philadelphia, PA, August 2005.

[10] K. Leibnitz, N. Wakamiya, and M. Murata, “Resilient multi-path routing based on a biological attractor selection scheme,” in *The Second International Workshop on Biologically Inspired Approaches to Advanced Information Technology (BioAdit 2006)*, Osaka, Japan, January 2006.

[11] ——, “Biologically inspired self-adaptive multi-path routing in overlay networks,” *Commun. ACM*, vol. 49, no. 3, pp. 62–67, 2006.

[12] J. Kingman, *Poisson Processes*, ser. Oxford Studies in Probability. New York: Oxford University Press, 1993, vol. 3.