

Adaptive-SDR: Adaptive Swarm-based Distributed Routing

I. Kassabalidis*, M.A. El-Sharkawi*, R.J.Marks II*, P. Arabshahi†, A.A. Gray†

* Dept. of Electrical Eng., Box 352500
University of Washington
Seattle, WA 98195 USA
[ikas,elsharkawi,marks]@ee.washington.edu

†Jet Propulsion Laboratory
4800 Oak Grove Drive, MS 238-343
Pasadena, CA 91109 USA
[payman,gray]@jpl.nasa.gov

Abstract – Swarm intelligence forms the core of a new class of algorithms inspired by the social behavior of insects that live in swarms. Its attractive features include adaptation, robustness and a distributed, decentralized nature, rendering swarm-based algorithms well-suited for routing in wireless or satellite networks, where it is difficult to implement centralized network control. We propose one such routing algorithm, dubbed *Adaptive Swarm-based Distributed Routing (Adaptive-SDR)*, which is scalable, robust and suitable to handle large amounts of network traffic, while minimizing delay and packet loss.

I. INTRODUCTION

Satellite networks exhibit characteristics of wireless and wired networks. Transmission is wireless but highly directional, therefore resembling a wired link but with higher delay. The ideal form of routing in any network is that which instantly responds to varying network conditions. However, this presents challenges in satellite networks, since the delays inherent in the network make centralized control inefficient. Thus, a distributed routing scheme is necessary to address these issues.

Swarm intelligence is an evolving field yielding methods for distributed systems optimization [1]. For the purpose of network routing, the paradigm that has already been proposed as most promising is that of ants. These remarkable insects have the ability to discover routes to their food sources through simple primitive interactions, via a chemical substance called a *pheromone*. Pheromones are used for indirect communication between ants, a phenomenon known as *stigmergy*. Its end effect is the manifestation of behavior far more complex than expected by such simple organisms. There already exist many successful adaptations of ant behavior to network control [1-11], with the most prominent being AntNet [2,3], and Ant-based Control (ABC) [4]. An overview of swarm-based algorithms can be found in [12]. The application of AntNet to a satellite network is discussed in [13].

These and the other existing algorithms lack a number of important features, namely scalability, full utilization of the network capacity, routing oscillations, and routing loops.

We propose an algorithm called *Adaptive Swarm-based Distributed (Adaptive-SDR)* that addresses these issues.

II. SWARM ROUTING TECHNIQUES

In all the above swarm-based algorithms [1-11], every node has a table of next-hop probabilities to each destination, as shown in Table. I.

Table I. Swarm-based Routing Table

		Next Hop	
		3	4
Destination Node	1	0.15	0.85
	2	0.75	0.25

Each row in this table corresponds to a destination and each column to a neighbor. The entries in the table are the probabilities that the next-hop is a specific neighbor. In previous swarm-based algorithms, these probabilities are used by ants to allow them to randomly explore the network and possibly find new and better routes. Then, once the routes are discovered, the next-hop probabilities are updated to reflect the new discoveries. As far as data packets are concerned, they are routed to the next-hop with the highest probability. A variation of this is cited in the literature [13], where data packets are routed in the same random manner as ants.

III. Adaptive-SDR

The algorithm consists of three parts. The first part is clustering the network nodes into colonies. The second part is finding network routes using special agents called ants. The third part is forwarding the network traffic using the routes discovered by the ants. The first part is not performed very frequently, just in the beginning stage of the algorithm and whenever the network topology changes enough to justify a re-clustering of the nodes. The second and third parts are performed constantly as part of regular network operation.

A. Clustering into Colonies

AntNet and ABC have scalability problems, since each node has to send an ant to all the other nodes of the network, which means that the total number of ants that have to be sent is $N \times (N-1)$. For large networks, the amount of traffic generated by the ants would be prohibitive. Furthermore, for distant destinations there is a larger likelihood of the ants being lost. Moreover, the large traveling times of the ant render the information they carry outdated.

One way of addressing all these issues is clustering the nodes into colonies. However, the optimal number of colonies is not obvious, but a simple derivation may be performed if the goal is to minimize the number of ants. Assuming that all nodes send ants at the same rate and that all colonies have approximately the same number of nodes, then the total number of ants is defined by:

$$\#ants = NN \left(\frac{NN}{NC} - 1 + NC - 1 \right) \quad (1)$$

where NN is the number of nodes in the network and NC is the number of colonies. The value of NC minimizing the number of ants is $NC = \sqrt{NN}$.

From the available techniques in the literature, we choose *k-means clustering* [14], where the distance function is the Euclidean distance between the nodes. Clustering is performed by a central controlling entity that is aware of the geographical locations of all the nodes.

B. Discovering Routes

After the network colonies have been formed, two types of agents, called ants, are introduced into the network. The first type is *colony ants* and the second type is *local ants*. The task of the colony ants is to find routes from one cluster to the other, while local ants are confined within a colony and are responsible for finding routes within their colonies. In order to facilitate this scheme, every node is equipped with two routing tables instead of one. The first, shown in Table II, is the colony routing table, while the second, shown in Table III is the local routing table.

Table II. Colony Routing Table

		Next Hop		
		3	4	5
Destination	A	0.1	0.05	0.85
Colony	B	0.2	0.55	0.25

Table III. Local Routing Table

		Next Hop	
		3	4
Destination Node	1	0.15	0.85
(within colony)	2	0.75	0.25

In the colony routing table, all the neighbors of the nodes are included in the next-hop list, while in the local routing table only the nodes that belong to the same colony are included as possible next-hops.

The probabilities of the routing tables are updated in a manner that borrows some features from the AntNet algorithm [2,3], while adding significant improvements.

1. At regular intervals, each node sends one local ant to every destination within the colony, and only one colony ant for each outside colony.

2. The local ants choose next-hops according to the local table probabilities, while the colony ants choose next-hops according to the colony tables.

3. Both types of ants record the nodes they follow and the corresponding arrival times.

4. An ant tries to avoid all previously visited nodes. If this is impossible, then a loop is detected and all the memory pertaining to the loop is erased and a random next hop is taken. Also, ants exceeding a certain number of hops to reach the destination are terminated. This number is usually set equal to the number of nodes in the network.

5. Once the destination is reached, the ants trace the path backwards to their source. When they arrive at each intermediate node, they update the routing tables' probabilities of that node.

The rules for updating the probabilities of both tables are the same. First, the remaining trip time T to the destination is calculated as follows.

$$T = t_{dst} - t_n, \quad (3)$$

where t_n is the arrival time at current node and t_{dst} the arrival at the destination. Next, the intermediate quantity λ is calculated as

$$\lambda = \begin{cases} \frac{T}{c\mu}, & c \geq 1, \text{ if } \frac{T}{c\mu} < 1, \\ 1, & \text{otherwise} \end{cases} \quad (4)$$

where μ is the mean of T of all the ants that have passed from this node and had the same destination and c is a scaling factor usually set by the user, normally equal to 2. Then, λ is used to calculate the changes of the probabilities as follows.

$$\begin{aligned} d_{next} &= \alpha \times (1 - \lambda) (1 - P_{n,dst}) + w * \Delta P_{n,dst} \\ d_{prev} &= -\alpha \times (1 - \lambda) P_{p,dst} + w * \Delta P_{p,dst} \\ d_{rest} &= -(1 - \lambda) P_{r,dst} + w * \Delta P_{r,dst} \\ r \neq n, r \neq p, r \in N_k \end{aligned} \quad (5)$$

where d_{next} , d_{prev} , d_{rest} and $P_{n,dst}$, $P_{p,dst}$ and $P_{r,dst}$ are the step sizes and the probabilities of the next-hop, the previous-hop and the rest of the neighbors, respectively. N_k is the set of neighbors of the current node k and $\Delta P_{n,dst}$, $\Delta P_{p,dst}$ and $\Delta P_{r,dst}$ are the previous changes in the probabilities of the next-hop, the previous-hop and the rest of the neighbors. α and w are scaling and inertia factors.

The function of the scaling factor α is to prevent network oscillations by limiting changes in the probabilities. The inertia factor w has the same effect by providing inertia to the changes in the probabilities.

Finally, the routing table probabilities are updated as follows:

$$\begin{aligned} P_{n,dst} &= P_{n,dst} + d_{next} \\ P_{p,dst} &= P_{p,dst} + d_{prev} \\ P_{r,dst} &= P_{r,dst} + d_{rest} \end{aligned} \quad (6)$$

In Eq. (6), the next-hop receives a reward (positive d) so that the probability of that node being chosen by future ants passing through the current node is higher. The previous-hop receives a penalty (negative d), so that the likelihood of taking that next-hop by future ants passing through the current node is decreased. Finally, the probabilities of the rest of the nodes are decreased.

Next, all the probabilities are thresholded between $\frac{1}{(\#neighbors)^2}$ and 0.8. The reason for the thresholding is to prevent saturation of the probabilities that prevent exploration by the ants. The lower threshold is set at that value so that the ratio of the initial probabilities and the lowest possible

probabilities is equal to the number of neighbors. The higher threshold is set to this value after sufficient experimentation. Since all the above operations yield probabilities with sum not equal to unity, the probabilities are normalized so that the unity sum condition is met.

C. Traffic forwarding

The routing tables derived by the random movements of ants provide an estimate of the condition of the network. As far as data packets are concerned, the next-hop with the highest probability represents the node with the best promise to lead to the destination. Therefore, in algorithms like AntNet or ABC, data packets are routed to the node with the highest probability. This does not utilize the full network capacity. A variation of this is routing data packets in the same random manner as ants [13]. This leads to higher network utilization. However, the advantage of doing this is not clear, since ants are known to follow paths that sometimes are far worse than optimal as part of their exploration tasks. This is fine for small ant packets, but not for data. It is therefore necessary to take the present network conditions into account.

A first-order approach to network condition-sensitive routing is to monitor the state of the queues of the links to all outgoing neighbors and adapt the next-hop probabilities according to the load in each queue. Then, the next-hop with the highest adjusted probability is selected. A simple measure of the state of a queue is the total size of its packets. The probabilities of the routing table can then be temporarily adjusted as follows:

$$P'_{i,dst} = P_{i,dst} \times \frac{1}{1 + load_i}, i \in N \quad (7)$$

where N is the set of neighbors of the node and $load_i$ is the load in the queue of the link from the current node to node i .

This adjustment is only temporary and for the purpose of forwarding data at that specific time instant. It does not permanently change the routing table probabilities. The advantage of this method is that the best next-hop is penalized when it is congested, while it is still the best, and thus always chosen, when there is no congestion.

The data is forwarded to the destination using the following procedure:

1. When a data packet originates from the source, it knows the destination node and colony.
2. If the node where it is at does not belong to the same colony as the destination node, then the packet uses the next-hop with the highest probability of the colony routing table, as adjusted in Eq. (7). Otherwise;
3. If the packet is at a node that belongs to the same colony as the destination node, then the packet uses the next-hop with the highest probability of the local routing table, as adjusted by Eq. (7).

IV. TEST SYSTEM

We implement our algorithm on 4x4 and 7x7 networks as a proof of concept. The two network topologies are given in Figs. 1 and 2.

The 16-node network is divided into 4 colonies, while the 49-node network is divided into 7 colonies. The nodes are placed on 4x4 and 7x7 grids with random perturbations around the fixed grid positions. The dimensions of the grids for both cases are $10^7 \times 10^7$ m and the links' bandwidth is fixed at 1Mbps, while the delay is proportional to the distance between the nodes. All edges show bi-directional links. In the 16-node network there are 4 constant bit rate (CBR) sources, while the 49-node network has 7 CBR sources.

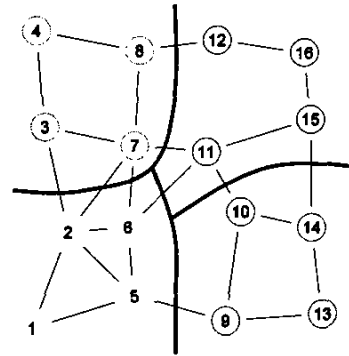


Fig 1. 16-node network divided into 4 colonies

The rate of the sources is set at 800kbps for the 16-node network and 960kbps for the 49-node network. The packet size is 1000bytes for the 16-nodes and 1200bytes for the 49 nodes. The interval is 0.01s for either network, which brings the offered load close to the capacity of the links. The traffic is intentionally high to test the algorithms' performance when the network capacity is stretched.

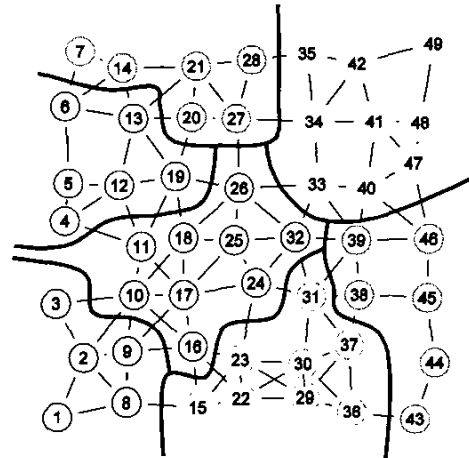


Fig 2. 49-node network divided into 7 colonies

We compare our algorithm to

- a) The original version of AntNet
- b) The version of AntNet with random data forwarding
- c) link-state algorithm (LS) [15]
- d) distance-vector algorithm (DV) [15].

The versions of the link-state and distance vector algorithms are the ones used by the popular simulation software "NS2" [16]. The results for the 16-node network are given in Table VII and in Fig. 3. For the 49-node network we only show results for Adaptive-SDR.

Table VII. Results for 16-node network

	Adaptive-SDR	AntNet	AntNet-Random	LS	DV
Delay(s)	0.3319	0.2022	0.2367	0.2010	0.2010
Throughput (Mbps)	2.672320	2.059800	1.561200	2.050200	2.050200
Packet loss ratio	0.1649	0.3563	0.4879	0.3593	0.3593

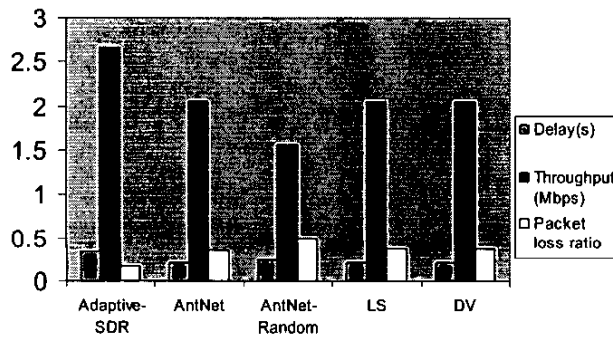


Fig. 3. Comparison of different routing algorithms on 16-node network

From the results in Table VII and Fig. 3 we notice the significant superiority of Adaptive-SDR based on packet loss and throughput. The average delay is higher for Adaptive-SDR, but this is due to the fact that slower routes are chosen when the faster ones are occupied. This is done to reduce packet loss.

Table VIII. Results for 49-node network

	Adaptive-SDR
Delay (s)	0.5825
Throughput (Mbps)	5.149800
Packet loss ratio	0.0804

The results of Table VIII show that Adaptive-SDR also performs well on larger systems. Comparison of Adaptive-SDR with other algorithms on larger real-life systems is part of ongoing research.

In this paper we have presented a novel algorithm based on swarm intelligence. The algorithm, dubbed Adaptive-SDR, has attractive features such as:

- 1) Scalability
- 2) High utilization of the network capacity
- 3) Avoiding routing oscillations and routing loops

Our proof of concept tests on the proposed algorithm have shown superior performance compared to other standard and swarm-based techniques.

VIII. ACKNOWLEDGEMENTS

The research described in this publication was carried out as part of a task funded by the Advanced Information Systems Technology (AIST) program at the NASA Office of Earth Science.

REFERENCES

- [1] E. Bonabeau, M. Dorigo, and G. Théraulaz, *Swarm intelligence: from natural to artificial systems*, Oxford University Press, 1999.
- [2] G. Di Caro and M. Dorigo, "AntNet: distributed stigmergetic control for communications networks," *Journal of Artificial Intelligence Research*, vol. 9, pp. 317-365, 1998.
- [3] G. Di Caro and M. Dorigo, "AntNet: a mobile agents approach to adaptive routing," Tech. Rep. IRIDIA/97-12, Université Libre de Bruxelles, Belgium
- [4] R. Schoonderwoerd, O. Holland, J. Bruten and L. Rothkrantz. "Ant-based Load Balancing in Telecommunication Networks," *Adaptive Behavior*, vol. 5, pp. 169-207, 1996.
- [5] A. Bieszczad, B. Pagurek, and T. White, "Mobile agents for network management", *IEEE Communication Surveys*, Fourth Quarter 1998, vol. 1, no. 1, 1998.
- [6] S. Lipperts and B. Kreller, "Mobile agents in telecommunications networks - a simulative approach to load balancing", *Proc. 5th Intl. Conf. Information Systems, Analysis and Synthesis, ISAS'99*, 1999.
- [7] M. Heusse, D. Snyers, S. Guérin, and P. Kuntz, "Adaptive agent-driven routing and load balancing in communication network", *Proc. ANTS'98, First International Workshop on Ant Colony Optimization*, Brussels, Belgium, October 15-16, 1998.
- [8] G. Di Caro and M. Dorigo, "Extending AntNet for best effort quality-of-service routing", *Proc. ANTS'98 - First International Workshop on Ant Colony Optimization*, Brussels, Belgium, October 15-16, 1998.
- [9] G. Navarro Varela and M.C. Sinclair, "Ant colony optimization for virtual-wavelength-path routing and wavelength allocation", *Proc. 1999 Congress on Evolutionary Computation*, Washington DC, USA, pp. 1809-1816, July 1999.
- [10] White T., Pagurek B., and Oppacher, F., Connection Management using Adaptive Agents. In Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'98), July 12th-16th, 1998, pp. 802-809.
- [11] E. Bonabeau, F. Henaux, S. Guerin, D. Snyers, P. Kuntz, and G. Théraulaz, "Routing in telecommunications networks with "smart" ant-like agents", *Proc. Intelligent Agents for Telecommunications Applications '98*.
- [12] I. Kassabalidis, M.A. El-Sharkawi, R.J. Marks II, P. Arabshahi, and A.A. Gray, "Swarm intelligence for routing in communication networks," *IEEE Globecom 2001*, Nov 25-29, 2001, San Antonio, Texas.
- [13] E. Sigel, B. Denby, and S. Le Hégarat-Mascle, "Application of ant colony optimization to adaptive routing in a LEO telecommunications satellite network," submitted to *IEEE Transactions on Networking*, July, 2000.
- [14] S. Haykin, *Neural Networks, A comprehensive foundation*, Prentice Hall, 1994.
- [15] D. Bertsekas and R. Gallager, *Data Networks*, Prentice-Hall, Inc, Upper Saddle River, New Jersey, 1992.
- [16] <http://www.isi.edu/nsnam/ns>