

Termite: Ad-Hoc Networking with Stigmergy

Martin Roth and Stephen Wicker
 Wireless Intelligent Systems Laboratory
 School of Electrical and Computer Engineering
 Cornell University
 Ithaca, New York 14850 USA
 {roth, wicker}@ece.cornell.edu

Abstract—A biologically inspired algorithm is presented to route messages in mobile wireless ad-hoc networks. The principles of swarm intelligence are used to define a probabilistic algorithm for which routing through paths of maximum throughput is an emergent property. This adaptive algorithm, dubbed *Termite*, uses stigmergy to reduce the amount of control traffic needed to maintain a high data goodput. Stigmergy is a process by which information is indirectly communicated between individuals through their environment. The Termite environment is the contents of all routing tables. The movement of packets is influenced at each node, and communicating nodes observe this influence to update their own tables. Strong routing robustness is achieved through the use of multiple paths; each packet is routed randomly and independently.

I. INTRODUCTION

A recent trend in mobile wireless ad-hoc networking (MANET) has been to strengthen existing approaches by considering more detailed network properties. Early schemes sought to adapt only to the network topology, such as finding a shortest hop path or perhaps a minimum energy path. However, a MANET environment is affected by many more factors than simply changes in topology. Additional factors may include traffic congestion, link quality and variability, relative node mobility and local topological stability, or the effects of a specific medium access (MAC) layer protocol.

A biologically inspired approach is proposed to adapt to the aggregate effects of each of these phenomena by finding paths of maximum throughput [7]. Furthermore, the proposal maintains strong routing robustness and low control traffic overhead.

A social insect metaphor suggests a probabilistic routing algorithm. Information about the network environment, including topology, link quality, traffic congestion, etc., is determined from the arrival rate of packets at each node. Packets are considered to route themselves and are able to influence the paths of others by updating routing parameters at each node. The collection of these parameters from all nodes across the network constitute the environment in which the packets exist. This Termite environment is a representation of the network environment. The interaction between packets and

their environment implicitly spreads information about network conditions and thus reduces the need to generate explicit control traffic. The method of communicating information indirectly through the environment is known as stigmergy.

A. Previous Work

A great deal of literature has been published in the field of routing in mobile ad-hoc networks. Early proposals optimize for traditional metrics such as path length or energy use [10] [9] [11] [12]. Proven techniques such as distance vector or link state algorithms are used to find optimal routes. Later efforts take advantage of more specific features of the network, such as physical layer effects or various link layer statistics [15] [16].

There exists relatively little work with regards to biologically inspired algorithms for routing in MANETs. A number of notable examples exist for wired networks, each showing a significant performance gain over traditional approaches. These schemes include Ant Based Control [4] and AntNet [1] [2]. In general, mobile agents travel the network while updating each visited node with routing information. Another major feature is the use of stigmergy.

The following paragraphs describe related work in ad-hoc routing.

a) Mobile Ants Based Routing: Mobile Ants Based Routing (MABR) is introduced as the first routing algorithm for MANETs inspired by social insects [3]. The approach presented in AntNet is extended to ad-hoc networks by abstracting the network into logical links and nodes based on relative node location. Location data is assumed from positioning devices. An optimized greedy routing algorithm is used to forward messages between logical nodes. No performance data is available for this protocol at the time of this writing.

b) Ant-Colony Based Routing Algorithm: This algorithm (ARA) presents a detailed routing scheme for MANETs, including route discovery and maintenance mechanisms [5] [6]. Route discovery is achieved by flooding forward ants to the destination while establishing reverse links to the source. A similar mechanism is employed other algorithms such as AODV. Routes are maintained primarily by data packets as they flow through the network. In the case of a route failure, an attempt is made to send the packet over an alternate link. Otherwise, it is returned to the previous hop for similar processing. A new route discovery sequence is launched if the packet is eventually returned to the source.

This research is sponsored by the Defense Advance Research Projects Agency (DARPA), and administered by the Army Research Office under Emergent Surveillance Plexus MURI Award No. DAAD19-01-1-0504. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

B. Structure of Paper

Section II describes the Termite routing algorithm. Section III is a discussion of the properties of Termite. Section IV details simulation results showing proof-of-concept. Section V notes the current state of the algorithm as well as the direction of future work. Section VI concludes the paper.

II. THE TERMITE ALGORITHM

Termite is a routing protocol for mobile wireless ad-hoc networks based on the principles of swarm intelligence [8]. This framework is used to define rules for each packet to follow, which result in emergent routing behavior. Reduced control traffic and quick route discovery and repair are additional benefits.

As packets are dispatched from a source to a destination, each follows a bias towards its destination while biasing its path in reverse towards its source. This bias is known as pheromone. Pheromone is layed on the communications links between nodes. Packets are attracted towards strong pheromone gradients but the next hop is always randomly decided. Following a destination pheromone trail while laying source pheromone along the same trail increases the likelihood of packets following that reverse path to the source. This is positive feedback. In order to prevent old routing solutions from remaining in the collective network memory, exponential pheromone decay is introduced as negative feedback. Pheromone increases linearly per packet, but decreases exponentially over time.

A. The Pheromone Table

Each node maintains a table tracking the amount of pheromone on each neighbor link. Each node has a distinct pheromone scent. The table may be visualized as a matrix with neighbor nodes listed along the side and destination nodes listed across the top. Rows correspond to neighbors and columns to destinations.

An entry in the pheromone table is referenced by $P_{n,d}$ where n is the neighbor index and d denotes the destination index. In other words, $P_{n,d}$ is the amount of pheromone from node d on the link with neighbor n .

1) *Table Size*: Termite's pheromone table is analogous to a routing table. The table can have up to N rows and D columns. The number of neighbors of a node is N , and D is the number of destinations in the network except for the node itself. Each cell in the table contains the amount of pheromone on the link to neighbor n from destination d . The table contains at most $N \cdot D$ entries, but its size ultimately depends on which destinations have been heard from, as well as the number of neighbors.

2) *Pheromone Update*: When a packet arrives at a node, the pheromone for the source of the packet is incremented by a constant, γ . The nominal value of γ is one. Only packets addressed to a node must be processed. A node is said to be addressed if it is the intended next hop recipient of the packet. Section Four describes an optional promiscuous mode. Equation (1) describes the pheromone update procedure when

a packet from source s is delivered from previous hop p . A prime indicates the updated value.

$$P'_{p,s} = P_{p,s} + \gamma \quad (1)$$

3) *Pheromone Decay*: To account for pheromone decay, each value in the pheromone table is periodically multiplied by the decay factor, $e^{-\tau}$. The decay rate is $\tau \geq 0$. A high decay rate will quickly reduce the amount of remaining pheromone, while a low value will degrade the pheromone slowly. The nominal pheromone decay interval is one second; this is called the decay period. Equation (2) describes pheromone decay.

$$P'_{n,d} = P_{n,d} \cdot e^{-\tau} \quad (2)$$

If all of the pheromone for a particular node decays, then the corresponding row and/or column is removed from the pheromone table. Removal of an entry from the pheromone table indicates that no packet has been received from that node in quite some time. It has likely become irrelevant and no route information must be maintained. A column (destination listing) is considered decayed if all of the pheromone in that column is equal to a minimum value. If that particular destination is also a neighbor then it cannot be removed unless all entries in the neighbor row are also decayed. A row is considered decayed if all of the pheromone values on the row are equal to the pheromone floor.

Neighbor nodes must be specially handled because they can forward packets as well as originate packets. A decayed column indicates that no traffic has been seen which was sourced by that node. However, a neighbor's role as traffic source may be secondary to its role as a traffic relay. The neighbor row must be declared decayed before the neighbor node can be removed from the pheromone table.

If a neighbor is determined to be lost by means of communications failure (the neighbor has left communications range), the neighbor row is simply removed from the pheromone table.

4) *Pheromone Bounds*: There are three values governing the bounds on pheromone in the table. These are the *pheromone ceiling*, the *pheromone floor*, and the *initial pheromone*. When a packet is received from an unknown source, a new entry for that node is created in the pheromone table. In the case of a neighbor node, a new column and row will be created (neighbor nodes are also potential destinations). If the source is not a neighbor, only a column is entered into the table. Each pheromone value in the new cells will be assigned the initial pheromone value. During the course of pheromone decay, no value is allowed to fall below the pheromone floor. This allows unused nodes to be easily detected. Likewise, no pheromone value is allowed to exceed the pheromone ceiling. These bounds prevent extreme differences in pheromone from upsetting the calculation of next hop probabilities.

B. Route Selection

Upon arrival to a node, b , an incoming packet with destination d is routed randomly based on the amount of d 's pheromone present on the neighbor links of b . A packet is

never forwarded to the same neighbor from which it was received, p . If b has only one neighbor, ie. the node that the packet was just received from, the packet is dropped. The equation below details the transformation of pheromone for d on link n , $P_{n,d}$, into the probability, $p_{n,d}$, that the packet will be forwarded to n . This is the forwarding function.

$$p_{n,d} = \frac{(P_{n,d} + K)^F}{\sum_{i=1}^N (P_{i,d} + K)^F} \quad (3)$$

The constants F and K are used to tune the routing behavior of Termite. The value of K determines the sensitivity of the probability calculations to small amounts of pheromone. If $K \geq 0$ is large, then large amounts of pheromone will have to be present before an appreciable effect will be seen in the routing probability. The nominal value of K is zero. Similarly, $F \geq 0$ may be used to modulate the differences between pheromone amounts. For example, $F > 1$ will accentuate differences between links, while $F < 1$ will deemphasize them. $F = 1$ yields a simple normalization. The nominal value of F is two.

C. Packet Design

There are five types of packets used by Termite. These are *data*, *hello*, *seed*, *route request (RREQ)*, and *route reply (RREP)*. The latter four types are control messages. Each packet type contains at least six fields, including *source address*, *destination address*, *previous hop address*, *next hop address*, *message identification*, and *Time-To-Live (TTL)*. Data packets may contain additional fields such as *data length* and *bulk data*.

1) *Data Packets*: Data packets are routed normally through the network. If a node does not know how to forward a packet, which is the case when the node's pheromone table does not contain the packet's destination, the packet is stored and a route request is issued. If a reply is not received within a given time period, *rreq timeout*, the data packet is dropped and considered lost.

2) *Route Request Packets*: Route request (RREQ) packets are sent when a node needs to find a path to an unknown destination. Route requests perform a random walk over the network until a node is found which contains some pheromone for the requested destination. In a random walk, a packet uniformly randomly chooses its next hop, except for the link it arrived on. If a route request cannot be forwarded, it is dropped. Any number of RREQ packets may be sent for each route request, the exact number of which may be tuned for a particular environment.

3) *Route Reply Packets*: Once a route request packet is received by a node containing pheromone to the requested destination, a route reply (RREP) packet is returned to the requestor. The RREP message is created such that the source of the packet appears to be the requested destination and the destination of the packet is the requestor. The reply packet extends pheromone for the requested destination back to the requestor without any need to change the way in which pheromone is recorded at each node. The reply packet is routed normally through the network by probabilistically following a pheromone trail to the requestor. Intermediate nodes on the return path automatically discover the requested node.

TABLE I
SIMULATION PARAMETERS

simulation area	50 x 50 [meters ²]
transmission range	10 [meters]
channel bit rate	1 [Mbps]
initial pheromone	1
pheromone ceiling	10000
pheromone floor	0.1
rreq timeout	2 [seconds]
τ (decay rate)	0.105
decay period	1 [second]
Data TTL	32 [hops]
RREQ TTL	32 [hops]
RREP TTL	32 [hops]
Seed TTL	4 [hops]
Seed period	30 [seconds]
Hello period	1 [second]
RREQs per Route Request	2

4) *Hello Packets*: Hello packets are used to search for neighbors when a node has become isolated. Hello packets are broadcast at a regular interval until a reply is received. A reply is sent by all nodes who hear the original hello. The node will stop sending hello packets until the pheromone table is again empty. Hello broadcasts may be avoided at the routing layer by an analogous mechanism at the MAC layer.

5) *Seed Packets*: Seed packets are used to actively spread a node's pheromone throughout the network. Seeds make a random walk through the network and serve to advertise a node's existence. They can be useful for reducing the necessary number of explicit route request transactions.

III. SIMULATION AND RESULTS

Termite has been simulated using Opnet Technologies Inc.'s, OPNET Modeler [14]. A series of tests are run in order to determine the viability of Termite as an effective scheme for MANET routing. Data goodput and control overhead are the primary metrics; node speed is the independent parameter.

A. Simulation Environment

Simulations are executed in scenarios containing one hundred nodes, lasting 600 seconds. Each node sends a packet with 64 bytes of data to a randomly chosen destination, at a constant rate of two per second. A perfect MAC layer model is used; a detailed IEEE 802.11b model is planned. Each scenario uses the simulation parameters as listed in Table One. Only node speed is varied and is indicated in the results. The random waypoint mobility model is used.

B. Simulation Results

In order to make an initial assessment of the algorithm, data goodput and control overhead are measured against average node speed. Results are shown in Figure One. Data goodput is the fraction of successfully delivered data packets. Overhead is measured in three ways; Bandwidth overhead is the fraction of all transmitted bits that belong to a control packet. Packet overhead is the fraction of all transmitted packets that are of the control type. Using these measures, packets transmitted multiple times will be counted in each instance. Control

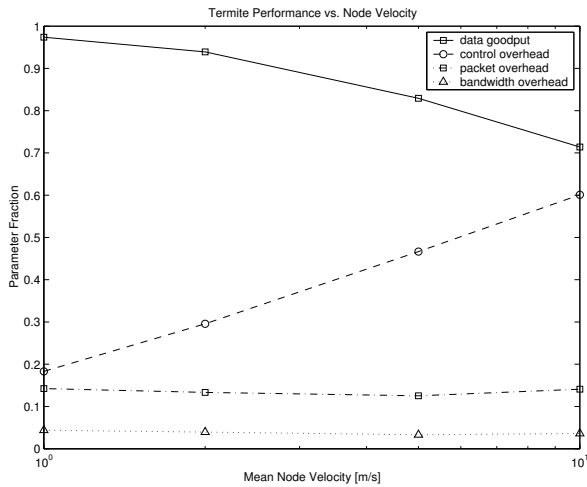


Fig. 1. Data Goodput vs. Control Overhead vs. Average Node Speed

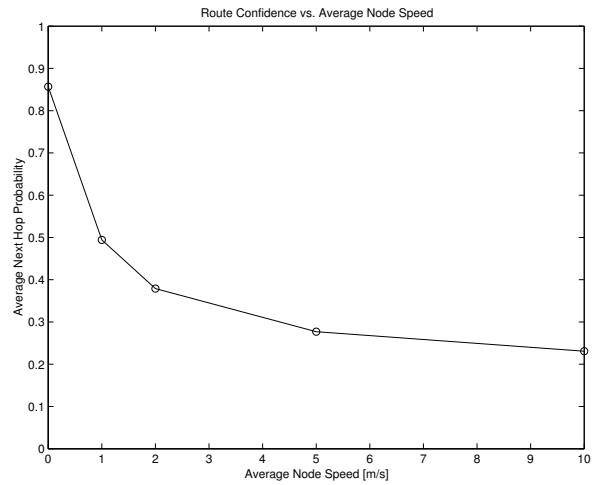


Fig. 3. Route Confidence vs. Mean Node Speed

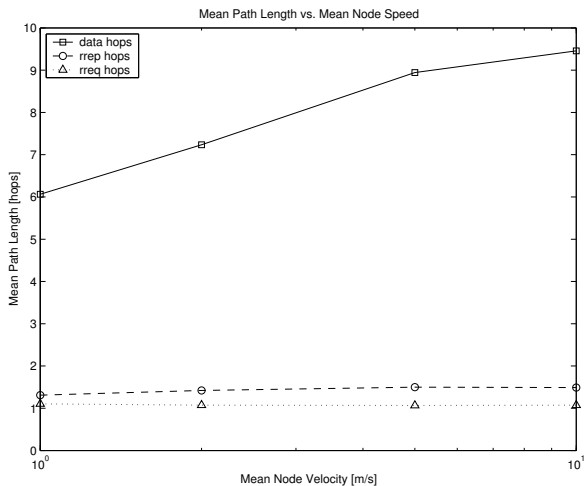


Fig. 2. Mean Packet Path Length vs. Mean Node Speed

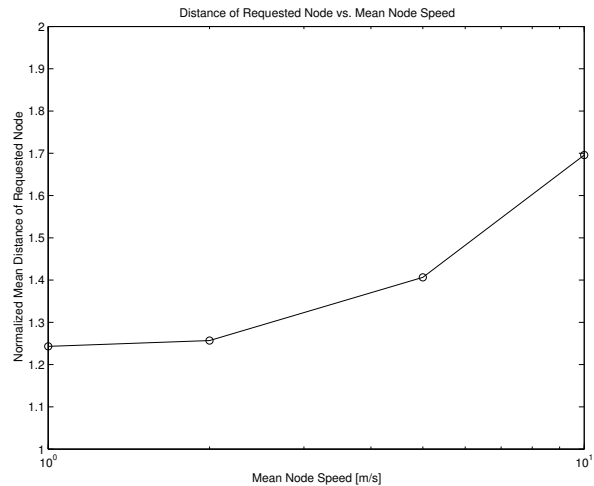


Fig. 4. Normalized Distance to Requested Destination vs. Mean Node Speed

overhead is the fraction of all sourced packets that are of the control type. Packets are only counted when they are created and not again.

Figure Two shows how the average hop count of data and route request packets varies with node mobility. The average path length of successfully delivered data packets experiences a sharp increase as the network becomes more volatile. However, the path length of both successful route request and route replies remains constant and quite low.

In order to more clearly illustrate how data packets take longer paths with higher node mobility, Figure Three shows how route confidence varies against average node speed. Route confidence is the average probability of the next hop. In the case of a static network, routes are extremely well established. The average next-hop probability drops as nodes increase their speed. Low probability hops yield longer paths as packets are more likely to wander through the network; they are no longer being closely guided to their destination.

Figure Four demonstrates the the distance of nodes being requested from the requestor, with respect to average node speed. The results are normalized to the transmission range,

which is ten meters in this simulation. The requested destinations tend to be within two hops, however higher node speeds quickly find the requested nodes to be farther away.

Finally, an indication of network volatility is shown in Table Two. Volatility is measured by the average number of link state changes per second per node. A change in link state is defined as an existing neighbor link failing, or a new neighbor link discovered. This metric observes the rate with which the network topology is changing from the individual node perspective. The increase in number of link changes is approximately linear with average node speed.

0 [m/s]	0 [link changes/s]
1	1.18
2	2.41
5	5.88
10	11.25

TABLE II
MEAN LINK STATE CHANGES PER SECOND PER NODE

C. Discussion

Figure One confirms that Termite is able to perform well over a variety of volatile network environments. Bandwidth overhead remains constant regardless of node mobility. This is despite the fact that control overhead can become quite large. Packet overhead mirrors the characteristics of bandwidth overhead although it is somewhat larger. This characteristic is in stark contrast to other MANET routing algorithms where overhead attributed to control packets is nontrivial in general and can become a dominating factor in network resource consumption [13]. Figure Two demonstrates that as the network becomes more volatile, data paths become longer while control paths remain small. The reason for this is explained by Figure Three, where data packets are quickly less confident in their next-hop selections as the network becomes unstable. Packets tend towards a random walk over the network as volatility increases.

Even though many more control packets may be generated in volatile situations, they do not have to travel very far to meet their objective. One explanation for this behavior draws on the results of Figure Four. The majority of requested destinations are within two hops of the requestor. This suggests that paths are being broken only at the very end of a trip. The destination moves outside of the communications range of the penultimate node, which then must issue a route request for it. The destination node has not moved very far and can be reached within two hops.

These results are promising; data goodput is maintained while packet overhead is kept constant, even in highly volatile networks. But there remain many open questions as to the specific operation of Termite. It is unclear what the optimal value of the system parameters are. How can these parameters be automatically determined based on locally available information. In these simulations, the parameters were chosen empirically and are not necessarily optimal. The optimal performance or even the optimal parameter values of this algorithm are unknown at this time.

IV. CURRENT STATUS AND FUTURE WORK

Current work focuses on simulating Termite. A variety of mobility and traffic scenarios are used in order to compare its performance against current state-of-the-art routing algorithms. Termite contains several tunable parameters, and methods are being found to automate the selection of optimal values according to network conditions. Of primary concern is the automatic determination of the decay rate for each node based on local information. This would include a single decay rate or a per neighbor decay rate. Other open questions include the effectiveness of seeding (proactive route maintenance), how often seed packets should be sent and how far, how many RREQ packets should be sent for each route request, or the need for Hello packets. Additionally, some consideration is being given to finding theoretical bounds on the performance of Termite, given its stochastic behavior. It is also planned to investigate the traffic load balancing properties in more detail. Future simulations will include a detailed MAC layer model.

V. CONCLUSION

A routing algorithm for mobile wireless ad-hoc networks has been presented. Swarm intelligence is used to build an emergent routing behavior. Packets probabilistically follow pheromone trails to their destination while biasing packets in the opposite direction. Passive route marking reduces the need for explicit routing traffic. Nodes determine network conditions by monitoring traffic flow and make adjustments to their routing tables.

Simulations show that Termite is able to maintain reasonable data goodput over a variety of mobility conditions. Control bandwidth overhead is minimized and remains constant across several degrees of network volatility.

REFERENCES

- [1] G. Di Caro, M. Dorigo, *Mobile Agents for Adaptive Routing*, Technical Report, IRIDIA/97-12, Universit Libre de Bruxelles, Belgium, 1997.
- [2] B. Barán, R. Sosa, *A New Approach for AntNet Routing*, Proceedings of the Ninth International Conference on Computer Communications and Networks, 2000.
- [3] M. Heissenbüttel, T. Braun, *Ants-Based Routing in Large Scale Mobile Ad-Hoc Networks*, Kommunikation in Verteilten Systemen (KiVS), 2003.
- [4] R. Schoonderwoerd, O. Holland, J. Bruten, L. Rothkrantz, *Ant-Based Load Balancing In Telecommunications Networks, Adaptive Behavior*, 1996.
- [5] M. Günes, U. Sorges, I. Bouazizi, *ARA - The Ant-Colony Based Routing Algorithm for MANETs* Proceedings of the ICPP Workshop on Ad Hoc Networks (IWAHN 2002), IEEE Computer Society Press, 2002, 79-85.
- [6] M. Günes, M. Kähler, I. Bouazizi, *Ant Routing Algorithm (ARA) for Mobile Multi-Hop Ad-Hoc Networks - New Features and Results*, The Second Mediterranean Workshop on Ad-Hoc Networks, 2003.
- [7] M. Roth, S. Wicker, *Termite: Emergent Ad-Hoc Networking*, The Second Mediterranean Workshop on Ad-Hoc Networks, 2003.
- [8] E. Bonabeau, M. Dorigo, G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, 1999.
- [9] C. Perkins, E. M. Royer, *Ad-hoc On-Demand Distance Vector Routing*, Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications, 1999.
- [10] D. Johnson, D. Maltz, *Dynamic Source Routing in Ad Hoc Wireless Networks*, Mobile Computing, edited by Tomasz Imielinski and Hank Korth (Kluwer Academic Publishers), chapter 5, 153-181, 1996.
- [11] P. Jacquet, P. Mühlethaler, T. Clausen, A. Laouiti, A. Qayyum, L. Viennot, *Optimized Link State Routing Protocol for Ad-Hoc Networks*, 2001.
- [12] J. Gomez, A. T. Campbell, M. Naghshineh, C. Bisdikian, *Conserving Transmission Power in Wireless Ad-Hoc Networks*, IEEE 9th International Conference on Network Protocols, 2001.
- [13] J. Broch, D. A. Maltz, D. B. Johnson, Y. Hu, J. Jetcheva, *A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols*, Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking, 1998.
- [14] <http://www.opnet.com/>
- [15] X.-H. Lin, Y.-K. Kwok, V. K. N. Lau, *BGCA: Bandwidth Guarded Channel Adaptive Routing for Ad-Hoc Networks*, Wireless Communications and Networking Conference, 2002.
- [16] C.-K. Toh, *A Novel Distributed Routing Protocol to Support Ad-Hoc Mobile Computing*, Conference Proceedings of the 1996 IEEE Fifteenth Annual International Phoenix Conference on Computer and Communications, 1996.