GPS/ANT-Like Routing in Ad Hoc Networks *

Daniel Câmara [a] Antonio A.F. Loureiro [b]

[a] *Department of Computer Science, Federal University of Minas Gerais, Caixa Postal 702, 30123-970 Belo Horizonte, MG, Brazil*
E-mail: danielc@dcc.ufmg.br
[b] *Department of Computer Science, Federal University of Minas Gerais, Caixa Postal 702, 30123-970 Belo Horizonte, MG, Brazil*
E-mail: loureiro@dcc.ufmg.br

A mobile ad hoc network (MANET) is comprised of mobile hosts that can communicate with each other using wireless links. In this paper we present a novel routing algorithm called GPSAL (GPS/Ant-Like Routing Algorithm) which is based on GPS (Global Positioning System) and mobile software agents modeled on ants for routing in ad hoc networks. We compare our algorithm to the Location-Aided Routing (LAR) [1] algorithm for MANET which is also based on GPS. Simulation results show that our algorithm has less overhead than LAR.

Keywords: MANET, Routing.

AMS Subject classification: TBD

1. Introduction

A mobile ad hoc network (MANET) is comprised of mobile hosts that can communicate with each other using wireless links. It is also possible to have access to some hosts in a fixed infrastructure depending on the kind of mobile ad hoc network available. Some scenarios where an ad hoc network could be used are business associates sharing information during a meeting, military personnel relaying tactical and other types of information in a battlefield, and emergency disaster relief personnel coordinating efforts after a natural disaster such as a hurricane, earthquake or flooding.

In this environment a route between two hosts may consist of hops through

* This work has been partially supported by Project SIAM/DCC/UFMG, grant MCT/FINEP/PRONEX number 76.97.1016.00

one or more nodes in the MANET. An important problem in a mobile ad hoc network is finding and maintaining routes since host mobility can cause topology changes. Several routing algorithms for MANETs have been proposed in the literature [1–13] and they differ in the way new routes are found and existing ones are modified.

Ant algorithms [14,15] have been developed recently and are based on a population approach, which has been successfully applied to several NP-hard combinatorial optimization problems [15–20]. As the name suggests, ant algorithms have been inspired by the behavior of real ant colonies, in particular, by their foraging behavior [21]. One of the main ideas of ant algorithms is the indirect communication of a colony of agents, based on pheromone trails. Pheromones are used by real ants for communication. The ants know the other ants paths by the pheromone trails, and the amount of pheromone on an trail reflects its importance. In GPSAL (GPS/Ant-Like Routing Algorithm) an ant agent has the responsibility of collecting and disseminating the information about the nodes' positions. In our case the software agents modeled on ants may follow different paths. In fact, the more different paths they follow the more the nodes' positions are disseminated. These software agents are implemented as a packet transmitted from node to node until the destination node is reached when it is sent back to the mobile unit that created it.

Most of the existing MANET routing algorithms do not consider the physical location of a destination node. In this paper we propose a novel routing algorithm called GPSAL which is based on GPS (Global Positioning System) [22,23] and mobile software agents modeled on ants for routing in ad hoc networks. Then we compare our algorithm to the Location-Aided Routing (LAR) [1] algorithm for MANET which is also based on GPS. Simulation results show that our algorithm has less overhead than the LAR algorithm.

This paper is organized as follows. Section 2 discusses the related work. In Section 3 we present the GPSAL algorithm and in Section 4 the algorithm is evaluated. Finally, in Section 5 we present some conclusions.

2. Related Work

The design of routing algorithms is a fundamental problem in a MANET and several routing protocols have been proposed in the literature [2–8,1,9–13]. An important property that a routing algorithm should have for this type of environment is the ability to adapt to different traffic patterns. This can be accomplished in various ways as explained below. A possible strategy is to consider the physical location of a destination node when choosing a route but this information is not taken into account in most of the existing routing algorithms for MANETs.

Johnson, Maltz and Broch [8] argue that the current routing protocols are inadequate for ad hoc mobile networks since the fraction of traffic related to routing information may be a significant portion of the wireless bandwidth available. This is the case in routing protocols that exchange periodically routing tables. Johnson, Maltz and Broch propose the algorithm DSR (*Dynamic Source Routing*) where each packet to be routed carries in its header a complete ordered list of nodes through which the packet must pass. This is a key aspect in the algorithm since intermediate nodes do not need to keep up-to-date routing information. The protocol is based on on-demand route discovery.

Several optimizations have been proposed to reduce the route discovery overhead. Perkins and Royer [12] propose the algorithm AODV (*Ad hoc On Demand Distance Vector routing*) that uses the on-demand mechanism of route discovery and route maintenance from DSR, and the hop-by-hop routing, sequence numbers and periodic beacons from DSDV [11].

Park and Corson [10] propose the algorithm TORA (*Temporally-Ordered Routing Algorithm*) that tries to minimize reaction to topological changes by limiting routing messages to the group of nodes near the change. In this scenario it is possible to have longer routes as a result of avoiding the overhead of discovering newer routes.

Hass and Pearlman [6] combine in the algorithm ZRP (*Zone Routing Protocol*) proactive and reactive approaches in the route discovery and route maintenance respectively. Route discovery is performed on-demand but limited to the initiator's neighborhood, and topology update propagation is limited to the neighborhood of the change.

The algorithms described above do not take into account the physical location of a destination node when choosing a route. Ko and Vaidya [1] propose the algorithm LAR (*Location-Aided Routing*) that uses the node's location when sending a packet to it. The basic idea is to perform a restricted flooding in the region where the destination node may be at the moment the initiator node performs a route discovery. The authors propose two variants of this algorithm. The first one, called LAR1, uses a request zone that has a rectangular shape. The zone size is defined to be the smallest rectangle that includes the current location of a host plus its expected zone. This value depends on the node's speed and the time elapsed since the last update. The second scheme is called LAR2 and includes in the routing packet the known location of a destination node and the expected distance the host may be from that location. This information is used by intermediate nodes to determine the request zone of a packet.

Basagni et al. [24] propose the algorithm DREAM (Distance Routing Effect Algorithm for Mobility) that is based on the principles of distance effect and mobility rate. Distance effect means that from the point of view of a given node the greater the distance to another node in the MANET, the slower they appear to be moving with respect to each other. The idea is to update the location information of the mobile nodes in the routing tables according to the distance separating them. Mobility rate is related to the frequency of location updates. In this way, each node sends updates to the other nodes in a frequency that depends on its degree of mobility.

Lin and Stojmenovic [25] also propose a routing algorithm based on GPS called GEDIR (*Geographic Distance Routing*) which deals only with static networks. Each node in the GEDIR algorithm chooses a neighbor that is closest to the destination in order to forward the message. Lin and Stojmenovic prove that directional based methods are not loop-free, while the GEDIR algorithm is. The algorithm does not worry about how a node learns and maintains the other nodes positions in its table, that is the greatest care of GPSAL. The delivery message of GEDIR, and its variants, are compared with various routing algorithms, including a modified LAR with interesting results. The main goal of GEDIR is the loop free routing.

In this paper we propose the algorithm GPSAL (*GPS/Ant-Like routing algorithm*) that is also based on the physical location of a destination node and mobile software agents modeled on ants. Ants are used to collect and disseminate information about nodes' location in the MANET.

Ant-like agents or the so-called mobile software agents were used by Appleby and Steward [26] for network control in telecommunications. Schoonderwoerd et al. [27] improved on the work of Appleby and Steward [26] by using simple agents that modify the routing tables of every node in a given network. Bonabeau et al. [28] proposed a simple mechanism based on ant-like agents for routing and load balancing in telecommunications based on the previous works of [26] and [27]. The results presented in [26–28] are very promising and the same principle is used in this paper.

3. GPSAL Algorithm

In this section we first describe how the location information is obtained in GPSAL, the routing table data structure, the role of fixed hosts, if available, in the MANET, the algorithm, and how it can be improved with the use of software agents modeled on ants.

3.1. Location Information

We assume that all mobile hosts participating in a MANET have a GPS unit which provides to the host its approximate three-dimensional position (latitude, longitude and altitude), velocity, and accurate time in Universal Time Coordinate (UTC) format. The proposed algorithm assumes that mobile hosts are moving in a plane, and therefore the altitude information is not used.

The information provided by a GPS unit may have some inaccuracy of a few meters depending on the system employed. In the algorithm proposed we assume that the location information is correct. The algorithm makes use of location information of a mobile host to reduce the number of routing messages.

3.2. Routing Table

All hosts in the MANET have a routing table where each entry represents a known host d and has the following information: current location of d, previous location of d, timestamp of current location, timestamp of previous location, and whether d is a mobile or fixed host. The timestamp field can be implemented as an integer value that is incremented monotonically.

The information in each entry may be updated whenever a host receives a more recent information about a given host d as we will see below.

3.3. Fixed Hosts

Depending on the MANET available it is possible to have access to some hosts in a fixed infrastructure. In our algorithm we assume that we may have access to fixed hosts. In Section 4 we compare the performance of the GPSAL algorithm when we vary the amount of fixed hosts available. In this case we are interested in evaluating hybrid networks comprised of both ad hoc and structured networks.

Often the cost to route packets in a fixed infrastructure is much less than in a MANET. In a traditional network it is common to have faster and more reliable links, and more powerful computers than in a mobile network. In case there is a route segment that passes through a fixed infrastructure we assume that its cost is insignificant and the wired network is able to find the nearest fixed host to the destination mobile computer.

3.4. Algorithm

Whenever a mobile node wants to join the MANET it listens to the medium to find out a neighbor node n. Once a neighbor node n is identified the mobile host sends a request packet to n asking for its routing table which is sent back to the host. From this moment on the new mobile host can start routing and sending packets in the MANET.

The routing protocol is based on the physical location of a destination host d stored in the routing table. If there is an entry in the routing table for host d, the best possible route is chosen using a shortest path algorithm. The route, comprised of a list of nodes and the corresponding timestamps, is attached to the packet which is sent to the first host in the list. If host d is not found in the routing table, the mobile node sends a message to the nearest fixed node, if available, that tries to find the destination node. Otherwise the data packet is not delivered.

Note that the information in the routing table, which was used to route the packet, probably reflects a snapshot in the past and the current network configuration may be different. Therefore each host, upon receiving a packet, compares the routing information present in the header with the information in its routing table. The entries that have older information than in the packet received are updated. This is performed by comparing the timestamp field in the packet received and in the routing table. Furthermore, each intermediate node can change the route to a destination node when there is a better route.

Figure 1 depicts two possible scenarios in a MANET when host A wants to send a packet to host G.

An important aspect of any routing algorithm for MANETs is how the routing table is updated. It is clear that better routes can be determined whenever a host has a more recent information about the network configuration. Routing information can be obtained both locally and globally. Local information is obtained from a neighbor node that periodically broadcasts only the changes occurred since the last time (this interval is a configuration parameter). Global information can be disseminated more rapidly using mobile software agents modeled on ants as explained in the following.

3.5. Disseminating Routing Information Using Ants

In a MANET when a mobile computer is powered on it may not know the physical location (current or past) of other hosts. However, this information can be gathered when a computer receives a message to be forwarded to another node or as a result of its own route discovery. All messages carry routing information that can be used by intermediate hosts in the routing process.

The route discovery can be accelerated using mobile software agents modeled on ants responsible for collecting and disseminating more up-to-date location information of mobile hosts. When a host receives an ant it compares the routing table present in the ant packet with its routing table and updates the entries that have older information as explained above. When this ant leaves a node it carries the most updated routing table from the point of view of the nodes already visited and the current one. This process is fundamental for the good performance of our algorithm. Of course there is an overhead associated with this process which can be controlled with the number of ants in the MANET.

Another important point is how to determine a destination to where an ant should be sent in order to collect more up-to-date information. One possibility is to choose the node with the oldest information in the routing table, i.e., the least recently entry in table to be updated, or the farthest distant node in the MANET from the current node, or just any node in the table. Whatever node is chosen, once it is reached the ant is sent back to the node that created it.

4. Performance Evaluation

In this section we evaluate different aspects of the GPSAL algorithm and also compare it to the Location-Aided Routing algorithm [1] which is also based on GPS. The simulations will be divided in two different sceneries. The first one shows the GPSAL algorithm, mainly the performance of mobile agents and the impact of the fixed network in the GPSAL, if present. The second one is an attempt to use the same environment presented in the original LAR paper [1]. The goal in this case is to make a fair comparison between the two algorithms.

In our simulations we follow the suggestions of performance issues and evaluation considerations presented in [29]. Our simulation environment treats packets similar to the IEEE 802.11 protocol [30,31], working in iterations. The algorithms were implemented in the C language and run in a Sun Sparc Ultra-1 with 128 MB of RAM and Sun Sparc Ultra-Enterprise with 512 MB of RAM, both running SunOS 5.5.1.

4.1. First Scenery

In this scenery all MANETs simulated have always 70 nodes including mobile and fixed hosts. The nodes are initially distributed in a $n \times n$ square region and each one knows its current location. We assume that each host moves continuously along the time with a maximum speed of $0.1n/iteration$. The simulation occurs in a discrete way, i.e., an iteration means that the algorithm executes in all nodes, and each message of the previous iteration was treated and the nodes can change their position. The direction of each host movement is chosen randomly. In case a host hits the grid border it bounces back and continues to move after reflection.

All nodes have the same transmission range that is $0.2n$. At any moment a node is disconnected from the MANET when in its transmission range there is no other mobile host.

In all simulations, the sender and destination nodes are chosen randomly. Data packets that cannot be delivered to a given destination due to a broken route are simply discarded. In our algorithm flooding is never used.

Simulation results shown in the next section are an average over 30 runs, each one with a different mobility pattern. When we compare the GPSAL algorithm to the LAR we use the same scenarios for both algorithms.

We do not take into account delays that may be introduced when multiple nodes in a neighborhood attempt to transmit simultaneously. We also do not consider transmission errors.

Updates in the routing tables are exchanged between hosts every three iterations. This value favors the LAR algorithm when we compare it to the GPSAL. A table exchange is the only way in the LAR to improve its knowledge of the network.

4.2. Results for First Scenery

The impact of introducing ants in routing dissemination is depicted in Figure 2 and can be further analyzed according to the data shown in Table 1. When a node converges it means that it has information about all nodes in the MANET. Figure 2 shows that for the first convergence the number of iterations decreases as the number of ants in the MANET increases. The amount of ants does not benefit the network once a network converges. The first line of Table 1, where the probability of each node to generate an ant is zero, for the GPSAL with ants, we use the routing information in the data packets and, therefore, those packets work as ants. For the version of GPSAL without ants we do not consider that piece of information.

Figure 3 shows the impact in the traffic when fixed hosts are used in the MANET. The traffic increases as the number of data packets between hosts also increases. As expected the number of traffic in the MANET decreases as the number of fixed hosts increases.

Figure 4 shows the effect of varying the amount of fixed nodes in the MANET for the GPSAL algorithm. In this scenario we consider the following parameters: 150 nodes, transmission range of 300 units and average speed of 4 units/iteration. In each iteration there is 0.5 probability of each node to send a data packet to another node in the network. In this case the routing overhead decreases as the amount of fixed nodes increases. It is interesting to note that the same overhead pattern is kept.

Figure 5 shows the overhead present in the GPSAL because of table exchanges and the introduction of ants. All tables are sent through broadcast. The value of "tables sent" means the amount of all routing information sent by all hosts whereas "tables received" means the amount of routing information received. As expected the overhead is greater when ants are introduced.

Finally in Table 2 we compare the amount of packets present in the MANET when one, two and three data packets are routed at the same time for both GPSAL and LAR (schemes 1 and 2). Since the LAR algorithm is based on restricted flooding the amount of packets increases exponentially as the number of concurrent data packets are being routed. This is a serious limitation of the LAR that is not present in GPSAL. This problem was also pointed out by Lin and Stojmenovic in [25]. The algorithm LAR2 uses a more restricted request zone and, therefore, the amount of data packet is decreased. We emphasize the types of packets in the comparisons. The GPSAL considers as traffic all packets (ants, data packets and table exchanges) of all nodes, even the nodes that did not participate in the data packet delivery. LAR considers only data packets generated by the data flooding. No branches are made in LAR and the timeout of packets is set to 10 hops.

4.3. Second Scenery

In the second simulation scenery we consider the number of nodes of the MANETS to be 15, 20, 25, 30, 35, 40, 45, and 50 for different simulation runs. The nodes are initially assigned a location in a 1000×1000 square region using a uniform distribution.

We assume that each node moves continuously along the time, without staying at any location, with an average speed uniformly distributed $v \pm 1.5$ units/iteration. We consider average speeds v in the range 1.5 to 32.5 units/iteration. The direction of movement of any node is chosen randomly. In case a node hits the grid border it bounces back and continues to move after reflection.

All nodes have the same transmission range. In the experiments, transmission ranges of 200, 250, 300, 350, 400, 450, and 500 units were considered. Two nodes cannot communicate with each other when they are beyond their transmission range and a node is disconnect from the MANET when in its transmission range there is no other mobile host.

In all simulations, there are 10 senders and one destination node chosen randomly. Data packets that cannot be delivered to a given destination due to a broken route are simply discarded. In our algorithm flooding is not used. Each sender generates on average 10 data packets/sec, with the time between two packets being exponentially distributed.

Updates in the routing tables are exchanged between hosts every three iterations. As mentioned before, this value favors the LAR algorithm when we compare it to the GPSAL.

We do not take into account delays that may be introduced when multiple nodes in a neighborhood attempt to transmit simultaneously. We also do not consider transmission errors.

4.4. Results for Second Scenery

Simulation results shown below are an average over 30 runs, each one with a different mobility pattern. For all runs, one of each input parameter, i.e., average speed, number of nodes, and transmission range, was varied and the others were kept constant. In the GPSAL algorithm, the probability of a node to generate an ant (as described in Section 3.5) in each iteration is 0.5.

For each input parameter, the result is presented as a function of number of routing packets per data packets. The term "data packets" (DP) refers to the number of data packets effectively received by the destination node and not the data packets transmitted by all senders. Recall that a data packet may be lost due to a broken route. The term "routing packets" (RP) refers to all routing related packets received by all nodes in the MANET. For the GPSAL algorithm this value also includes the overhead with ants.

When we compare the GPSAL algorithm to the LAR we use the same scenarios for both algorithms. However, we consider a simplification to the LAR algorithm. The first step in the LAR schemes is to perform a route discovery. If there is no reply within a timeout interval, the sender uses the flooding algorithm to find the route. In our simulations we do not consider this cost in the LAR algorithm since it is provided to the sender the location of the destination node.

Figures 6 and 7 show the effect of varying the transmission range. Typically, for the LAR schemes the routing overhead increases as transmission range and speed increase when there are several sources sending data packets to a destination node. With a larger transmission range, the number of routing packets increases since more nodes are reached outside of the request zone for the LAR algorithm. The GPSAL algorithm does not use flooding which provides a smaller overhead that also decreases as the speed increases. The variability in the curves is due to the combination of different scenarios where each one has 10 different senders and one destination. This fact can be seen from the curves of the GPSAL algorithm that restricts the transmission to its neighbors.

Figures 8 and 9 show the effect of varying the number of nodes in the MANET. The amount of routing overhead for both LAR schemes increases more quickly than in the GPSAL algorithm when the number of nodes in the MANET is increased. Considering a fixed transmission range of 300 units, the impact of increasing the speed and varying the number of nodes is much smaller than in the case of Figures 6 and 7.

Figure 10 shows the effect of varying the speed of nodes in the MANET. The amount of routing overhead for both LAR schemes has a similar pattern. LAR scheme 2 tends to perform better than LAR scheme 1.

5. Conclusions

In this paper we describe a novel routing protocol for MANETs that is based on mobile software agents modeled on ants. Ants are used to collect and disseminate information about the location of nodes in the network. This is a key aspect of the GPSAL algorithm that helps accelerate route discovery. Another feature of this algorithm is the use of fixed hosts whenever possible to route packets. The combination of these principles provide a better MANET routing algorithm.

Simulation results indicate that using location information helps in decreasing the routing overhead as compared to algorithms that do not use location information or resort to flooding. The GPSAL algorithm has a lower overhead when compared to the LAR protocol. We run several different experiments which show different trade-offs.

References

[1] Young-Bae Ko and Nitin H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. In *Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 66–75, Dallas, Texas, USA, October 25–30 1998.

[2] M.S. Corson and A. Ephremides. A distributed routing algorithm for mobile wireless networks. *ACM Journal on Wireless Networks*, 1(1):61–81, 1995.

[3] S. Corson, S. Batsell, and J. Macker. Architectural considerations for mobile mesh networking. Internet draft RPC, version 2, May 1996.

[4] B. Das, E. Sivakumar, and V. Bhargavan. Routing in ad-hoc networks using a spine. In *IEEE International Conference on Computer Communications and Networks*, 1997.

[5] R. Dube, D.D. Rais, K. Wang, and S.K. Tripathi. Signal stability based adaptive routing (SSA) for ad hoc mobile networks. *IEEE Personal Communications*, February 1997.

[6] Z.J. Haas and M.R. Pearlman. The zone routing protocol (ZRP) for ad hoc networks. Internet-draft, August 1998.

[7] M. Jiang, J. Li, and Y.-C. Tay. Cluster based routing protocol (CBRP) functional specification. Internet Draft, August 1998.

[8] D. Johnson, D.A. Maltz, and J. Broch. The dynamic source routing protocol for mobile ad hoc networks. Internet Draft, March 1998.

[9] P. Krishna, M. Chatterjee, N.H. Vaidya, and D.K. Pradhan. A cluster-based approach for routing in ad hoc networks. In *USENIX Symposium on Location Independent and Mobile Computing*, April 1995.

[10] V.D. Park and M.S. Corson. Temporally-ordered routing algorithm (TORA) version 1 functional specification. Internet Draft, August 1998.

[11] C.E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *ACM SIGCOMM Symposium on Communication, Architectures and Protocols*, 1994.

[12] C.E. Perkins and E.M. Royer. Ad hoc on demand distance vector (AODV) routing. Internet Draft, August 1998.

[13] C.-K. Toh. A novel distributed routing protocol to support ad-hoc mobile computing. *Wireless Personal Communication*, January 1997.

[14] M. Dorigo, V. Maniezzo, and A. Colorni. Positive feedback as a search strategy. Technical Report 91-016, Dip. Elettronica, Politecnico di Milano, 1991.

[15] M. Dorigo, V. Maniezzo, and A. Colorni. Positive feedback as a search strategy. *IEEE Transactions on Systems, Man, and Cybernetics*, Part B, 26(1):29–42, 1996.

[16] G. Di Caro and M. Dorigo. Antnet: Distributed stigmergetic control for communications networks. *Journal of Artifcial Intelligence Research (JAIR)*, 9:371–365, December 1998.

[17] M. Dorigo and L.M. Gambardella. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transac- tions on Evolutionary Computation*, 1(1):53–66, 1997.

[18] L.M. Gambardella and M. Dorigo. Has-sop: Hybrid ant system for the sequential ordering problem. Technical Report IDSIA 11-97, IDSIA, 1997.

[19] V. Maniezzo. Exact and approximate nondeterministic tree-search pro- cedures for the quadratic assignment problem. Technical Report CSR 98-1, Scienze dell'Informazione, Universit a di Bologna, Sede di Cesena, 1998.

[20] R. Michel and M. Middendorf. An island based ant system with looka- head for the shortest common supersequence problem. In *In Proceedings of the Fifth International Conference on Parallel Problem Solving from Nature*, volume 1498, pages 692–708. Springer Verlag, 1997.

[21] Sttzle T. and M. Dorigo. Aco algorithms for the traveling salesman problem. Technical Report IRIDIA/99-3, IRIDIA, Universit Libre de Bruxelles, 1999.

[22] Iowa State University GPS Page. http://www.cnde.iastate.edu/gps.html.

[23] NAVSTAR GPS Operations. http://tycho.usno.navy.mil/gpsinfo.html.

[24] S. Basagni et al. A distance routing effect algorithm for mobility (DREAM). In *Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 76–84, Dallas, Texas, USA, October 25–30 1998.

[25] Xu Lin and Ivan Stojmenovic. Geographic distance routing in ad hoc wireless networks. Technical report, Computer Science, SITE, University of Ottawa, December 1998.

[26] S. Appleby and S. Steward. Mobile software agents for control in telecommunications network. *British Telecom Technology Journal*, 12:104–113, 1994.

[27] R. Schoonderwoerd, O. Holland, J Bruten, and L Rothkrantz. Ant-based load balancing in telecommunications networks. *Adaptive Behavior*, 5:169–207, 1997.

[28] Eric Bonabeau et al. Routing in telecommunications networks with "smart" ant-like agents. In *Proceedings of the Second International Workshop on Intelligent Agents for Telecommunication Applications*, 1998. Lecture Notes in Artificial Intelligence, Springer Verlag.

[29] S. Corson and J. Macker. Mobile ad hoc networking (MANET): Routing protocol performance issues and evaluation considerations. Request for Comments 2501, January 1999.

[30] Phil Belanger and Wim Diepstraten. MAC entity: MAC basic access mechanism privacy and access control. Technical report, IEEE 802.11, March 1996.

[31] Greg Ennis. 802.11 architecture. Technical report, IEEE 802.11, March 1996.

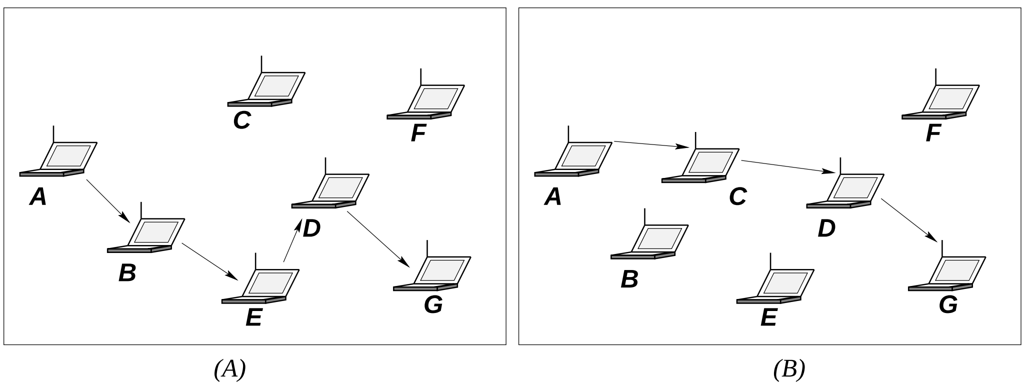

Figure 1. Host A wants to send a packet to host G and the initial route is shown on the left scenario. In the meantime host C changes its location as shown on the right scenario and a new route is chosen.

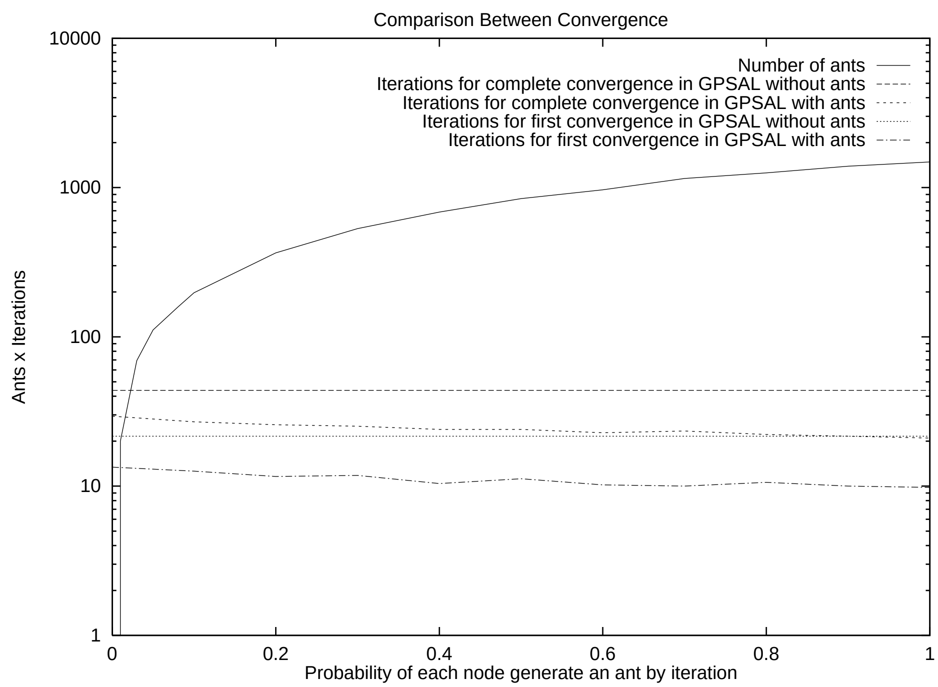

Figure 2. Convergence of the MANET using GPSAL

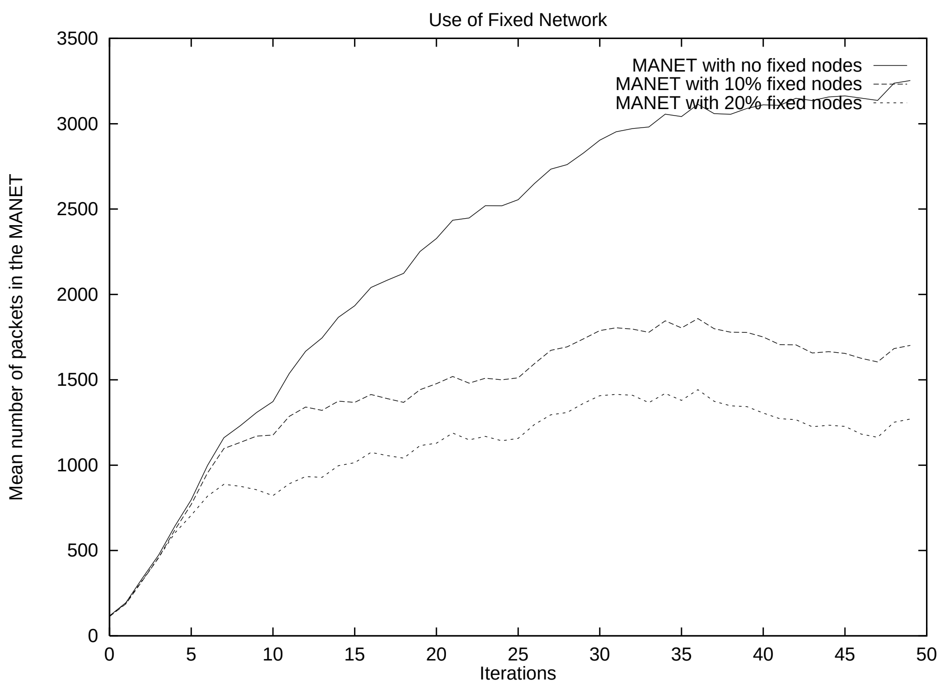

Figure 3. Traffic in the MANET using fixed hosts

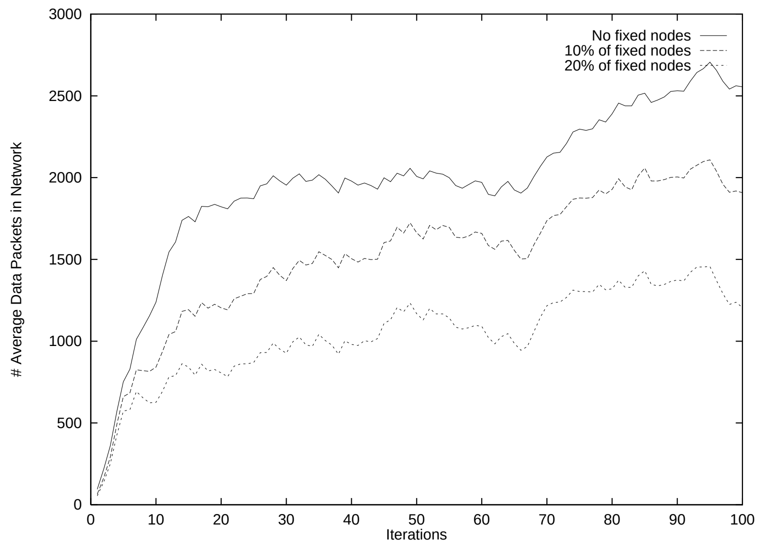

Figure 4. Impact of fixed nodes in the GPSAL algorithm.

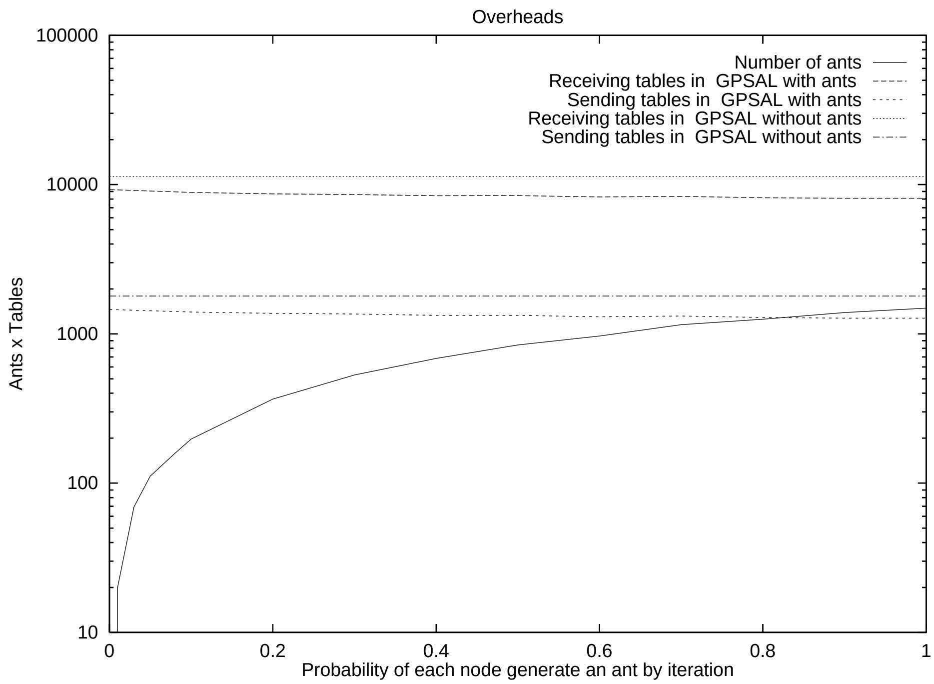

Figure 5. Overhead introduced with ants

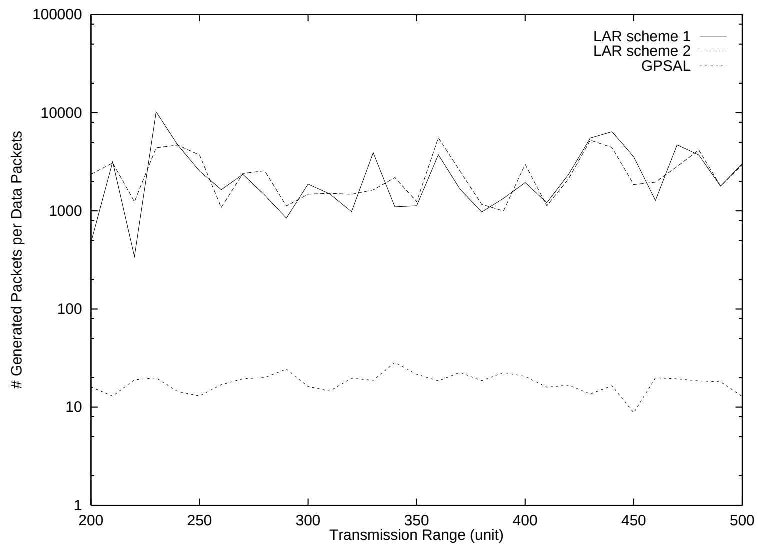

Figure 6. Transmission range versus number of RPs per DP considering a MANET with 30 nodes: Average speed of 9 units/iteration.

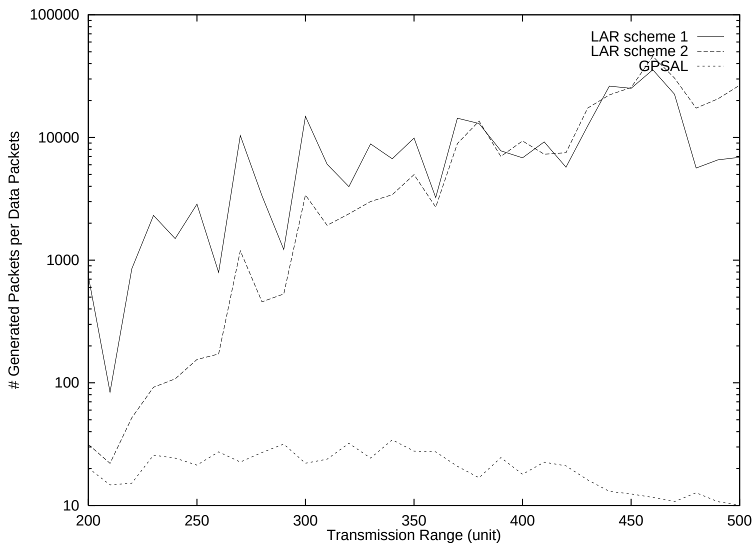

Figure 7. Transmission range versus number of RPs per DP considering a MANET with 30 nodes: Average speed of 50 units/iteration.

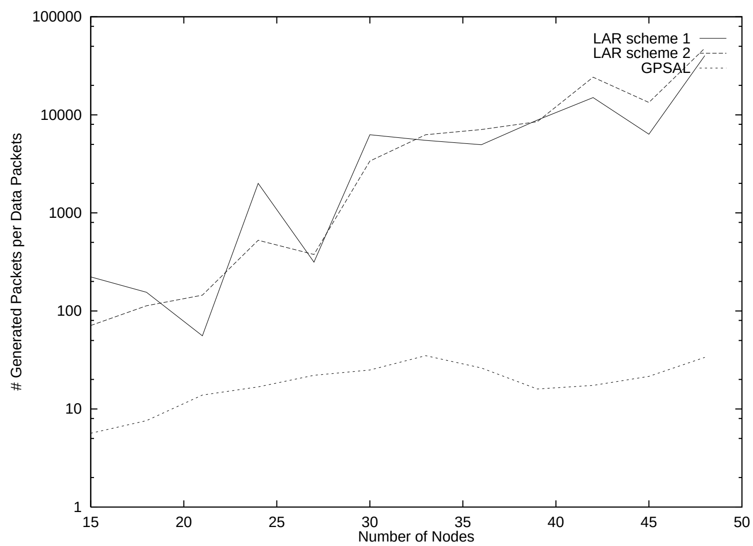

Figure 8. Number of nodes versus number of RPs per DP considering a MANET with transmission range of 300 units: Average speed of 9 units/iteration.

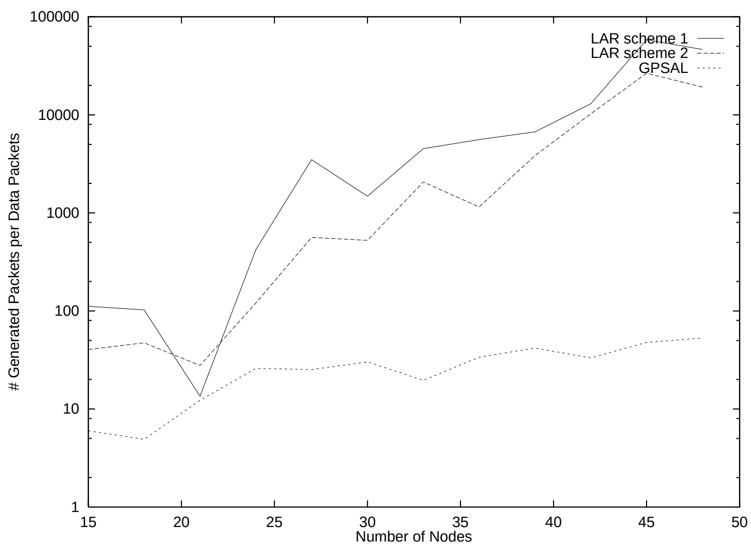

Figure 9. Number of nodes versus number of RPs per DP considering a MANET with transmission range of 300 units: Average speed of 50 units/iteration.

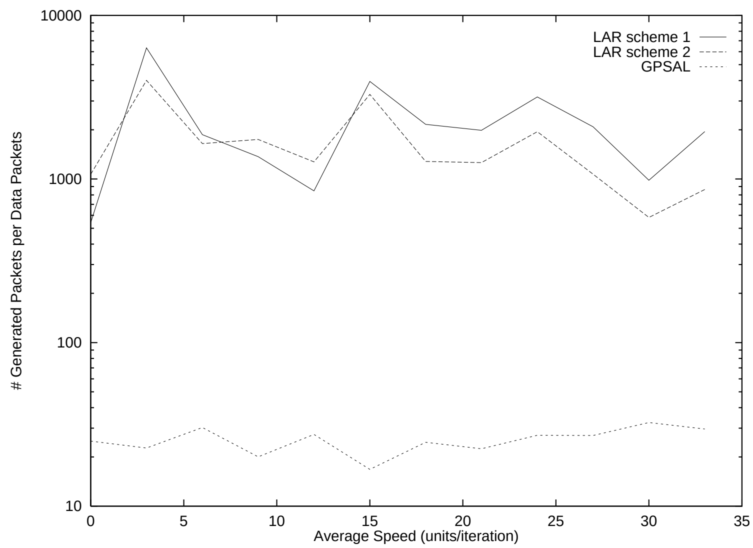

Figure 10. Average speed versus number of RPs per DP considering a MANET with transmission range of 300 units and 30 nodes.

Probability of each node to generate an ant by iteration	# of ants	# of iterations for the entire network convergence		# of iterations for the first network convergence	
		GPSAL without ants	GPSAL with ants	GPSAL without ants	GPSAL with ants
0.0	0	43.8	29.4	21.6	13.4
0.01	20.0	43.8		21.6	
0.03	69.3	43.8		21.6	
0.05	111.3	43.8		21.6	
0.08	158.0	43.8		21.6	
0.1	197.4	43.8	27.0	21.6	12.6
0.2	365.6	43.8	25.8	21.6	11.6
0.3	531.2	43.8	25.2	21.6	11.8
0.4	685.0	43.8	24.0	21.6	10.4
0.5	843.8	43.8	24.0	21.6	11.2
0.6	967.6	43.8	22.8	21.6	10.2
0.7	1151.8	43.8	23.4	21.6	10.0
0.8	1255.0	43.8	22.2	21.6	10.6
0.9	1391.0	43.8	21.6	21.6	10.0
1.0	1486.0	43.8	21.0	21.6	9.8

Table 1
Convergence of the MANET using GPSAL

	Data packets in the MANET		
	1	2	3
GPSAL	1430	2004	2133
LAR1	2 681 259	5 163 028	7 404 506
LAR2	113 603	2 315 741	8 154 490

Table 2
Amount of packets generated