

ANT-ROUTING-ALGORITHM (ARA) FOR MOBILE MULTI-HOP AD-HOC NETWORKS – NEW FEATURES AND RESULTS

Mesut Güneş, Martin Kähler, Imed Bouazizi

Department of Computer Science, Informatik 4
Ahornstr. 55, 52074 Aachen
University of Aachen, Germany
{mesut, mkaehmer, imed}@i4.informatik.rwth-aachen.de

ABSTRACT

A mobile ad-hoc network (MANET) is a collection of mobile nodes which communicate over radio. These networks have an important advantage, they do not require any existing infrastructure or central administration. Therefore, mobile ad-hoc networks are suitable for temporary communication links. This flexibility, however, comes at a price: communication is difficult to organize due to frequent topology changes. Ant algorithms are a class of swarm intelligence and try to map the solution capability of ant colonies to mathematical and engineering problems. The Ant-Colony-Based Routing Algorithm (ARA) is based on ant algorithms. The main properties of the algorithm are high adaptive, efficiency. In this paper we present some extensions to the basic idea and show through simulation results the performance gain and compare it with AODV and DSR.

1 INTRODUCTION

A mobile multi-hop ad-hoc network (MANET) is a set of mobile nodes which communicate over radio and do not require any infrastructure. These networks are very flexible and suitable for several types of applications, as they allow the establishment of temporary communication without any pre installed infrastructure (see figure 1). Due to the limited transmission range of wireless interfaces, in most cases communication has to be relayed via intermediate nodes. Thus, in mobile multi-hop ad-hoc networks each node

also has to be a router.

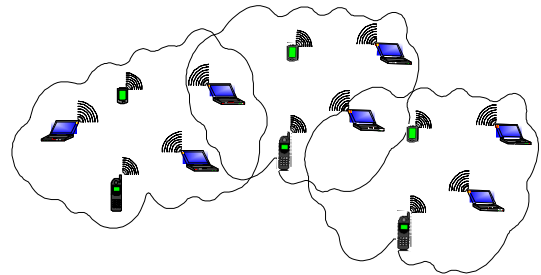


Figure 1: A mobile multi-hop ad-hoc network.

Beside the disaster and military application domain the deployment of mobile ad-hoc networks for multimedia applications is another interesting domain. However, the performance of such networks must be improved before this can be realized. With newly emerging radio technologies, e.g. IEEE 802.11a and Bluetooth, the realization of multimedia applications over mobile ad-hoc networks becomes more realistic.

To find a route between the communication endpoints is a major problem in mobile multi-hop ad-hoc networks. The problem is further aggravated through the node mobility. Many different approaches to handle this problem were proposed in recent years [1, 2], but so far no routing algorithm has been suitable for all situations. Other aspects of mobile ad-hoc networks are also subject to current research, especially the dynamic address configuration of nodes [3, 4, 5].

In this paper we present a new approach for an on-

demand ad-hoc routing algorithm, which is based on *ant algorithms*. Ant algorithms are a subset of swarm intelligence and try to mimic the ability of ants to solve complex problems through cooperation without direct communication. Several algorithms which are based on ant colony algorithms were introduced in recent years to solve, for instance, optimization problems.

The basic idea with some promising results was presented in [6]. In this paper we present some new extensions and show, through new results the performance gain compared to the basic idea. This concerns especially situations with high mobile nodes.

The remainder of this paper is organized as follows. In section 2 we present the basics of ant algorithms. Subsequently, in section 3 an overview of existing routing algorithms for mobile ad-hoc networks is given and related ant approaches are discussed. In section 4 we describe the routing algorithm ARA in detail and discuss its advantages, problems and the new extensions. In section 5 we present some simulation results to show the suitability of the approach and compare it with existing routing algorithms. Finally, a conclusion is given in section 6.

2 ANT ALGORITHMS

Ant algorithms are a subset of swarm intelligence which model the behavior of insect swarms to solve complex tasks by cooperation. They are multi-agent systems where agents show the behavior of individual ants. See [7, 8] for more information.

2.1 Ant foraging behavior

The basic idea of the ant algorithm is taken from the food searching behavior of real ants. When ants search for food, they start from their nest and walk toward the food. When an ant reaches an intersection, it has to decide which branch to take next. While walking ants deposit *pheromone* which marks the selected route. The concentration of pheromone on a certain path is an indication of its usage. Over time the concentration of pheromone decreases due to diffusion effects.

Figure 2 shows a scenario with two routes from the nest to the food. At the intersection the first ants randomly select a branch (see figure 2a)). Since the lower route is shorter than the upper one, the ants which take this path will reach the food place first. On their way back to the nest, the ants again have to select a path. After a while the pheromone concentration on the shorter path will be higher than on the longer path, because the ants using the shorter path will increase the pheromone concentration faster (see figure 2b)). Thus, eventually all ants will only use this path (see figure 2c)).

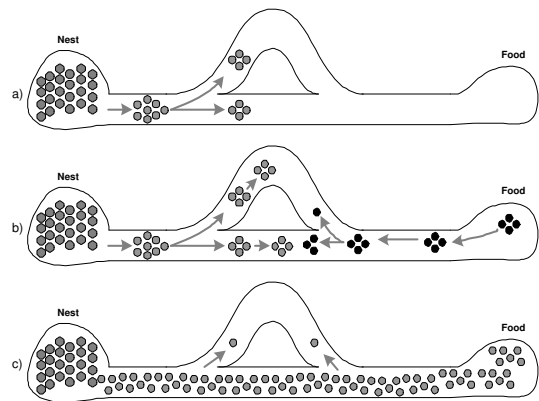


Figure 2: All ants take the shortest path after an initial searching time.

What happens if there exist a close dead end without food between the nest and the real food place (see figure 3)? One might assume the ants would amplify the path to the dead end because of their randomly motion. This is prohibited, since ants with food deposit more pheromone as ants without food. Hence paths leading to food is amplified much more stronger.

The foraging behavior of the ants can be used to find the shortest path in networks. Especially the dynamic component of this method provides for a high degree of adaptation to changes in mobile ad-hoc network topology, since in these networks the existence of links is not guaranteed and link changes occur frequently.

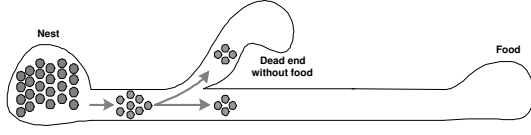


Figure 3: Example with near path with a dead end

2.2 A simple ant algorithm

Let $G = (V, E)$ be a connected graph with $n = |V|$ nodes. The simple ant colony optimization metaheuristic can be used to find the shortest path between a source node v_S and a destination node v_D on the graph G . The path length is given by the number of nodes on the path. A variable $\varphi_{i,j}$ (artificial pheromone), which is modified by the ants when they visit the node is associated with an edge $e(i, j) \in E$ of the graph connecting the nodes v_i and v_j . The pheromone concentration $\varphi_{i,j}$ is an indication of the usage of this edge. Initially $\varphi_{i,j}$ is constant for each edge $e(i, j)$.

An ant located in node v_i uses pheromone $\varphi_{i,j}$ of node $v_j \in N_i$ to compute the probability of node v_j being the next hop. N_i is the set of one-step neighbors of node v_i . The transition probabilities $p_{i,j}$ of a node v_i , i.e. the probability that the ant selects node v_j after it has visited v_i , are defined as follows

$$p_{i,j} = \begin{cases} \frac{\varphi_{i,j}}{\sum_{k \in N_i} \varphi_{i,k}} & \text{if } j \in N_i \\ 0 & \text{if } j \notin N_i \end{cases}, \quad \sum_{j \in N_i} p_{i,j} = 1.$$

During the route finding process, ants deposit pheromone on the edges. In the simplest version of the algorithm, the ants deposit a constant amount $\Delta\varphi$ of pheromone, i.e. the amount of pheromone of the edge $e(v_i, v_j)$ when the ant is moving from node v_i to node v_j is changed as follows:

$$\varphi_{i,j} := \varphi_{i,j} + \Delta\varphi$$

Hereby, the positive reinforcement is given. Like real pheromone the artificial pheromone concentration decreases with time which represents the negative reinforcement. In the simple ant algorithm this is described by:

$$\varphi_{i,j} := (1 - q) \cdot \varphi_{i,j}, \quad q \in (0, 1] \quad (1)$$

After forwarding a data packet the entries in the routing table of node v_i change as follows:

$$\varphi_{i,j} := \begin{cases} (1 - q) \cdot \varphi_{i,j} + \Delta\varphi & , \text{ if } j \text{ next node} \\ (1 - q) \cdot \varphi_{i,j} & , \text{ else} \end{cases}$$

2.3 Why ant algorithms are suitable for ad-hoc networks

The simple ant algorithm shown in the previous section illustrates different reasons why this kind of algorithms could perform well in mobile multi-hop ad-hoc networks. We discuss some by relating them to important properties of mobile ad-hoc networks.

- **Dynamic topology:** This property is responsible for the poor performance of many 'classical' routing algorithms in mobile multi-hop ad-hoc networks. The ant algorithm is based on autonomous agent systems imitating individual ants. This allows a high adaptation to the current topology of the network.
- **Local work:** In contrast to other routing approaches, the ant algorithm is based only on local information, i.e. no routing tables or other information blocks have to be transmitted to other nodes of the network.
- **Link quality:** It is possible to integrate the connection/link quality into the computation of the pheromone concentration, especially into the evaporation process. This will improve the decision process with respect to the link quality. It is important to note that the approach can be modified so that nodes can also manipulate the pheromone concentration independent of the ants, e.g. if a node detects a change of the link quality.
- **Support for multi-path:** Each node has a routing table with entries for all its neighbors which also contains the pheromone concentration. The decision rule for selection of the next node is based on the pheromone concentration at the current node which is provided for each possible link. Thus, the approach supports multi-path routing.

3 RELATED WORK

3.1 MANET Routing Algorithms

Dynamic Source Routing (DSR)

The basic idea of source routing is that the source node includes the full routing information into each data packet, e.g. $(v_S, v_1, v_2, \dots, v_D)$, for a packet to v_D which is routed via v_1, v_2 , etc.

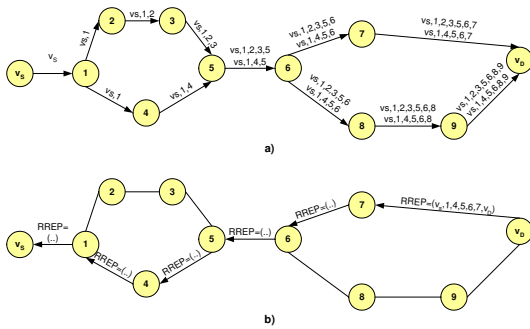


Figure 4: Route discovery in DSR. a) The source v_S broadcasts a RREQ which is forwarded to the destination v_D . b) The destination v_D sends the shortest source route to the source node in a RREP message. The number of hops is the metric used here.

The Dynamic Source Routing [9, 10, 1] (DSR) applies the method of source routing to mobile ad-hoc networks. The main question is how to obtain a source route for a certain destination. DSR uses two phases: i) route discovery, and ii) route maintenance.

If a source node v_S does not have a route for a certain destination v_D it initiates a route discovery process by broadcasting a route request RREQ to its neighbors. The RREQ is a small packet containing v_S , v_D , a unique id RREQ_ID, and L_{SD} which is the list of nodes which forwarded the RREQ. An intermediate node receiving a RREQ for the first time appends its address to L_{SD} and broadcasts it to its neighbors, but not back to the node where the request came from.

If the destination node v_D receives the route request it extracts L_{SD} , creates a route reply message RREP containing L_{SD} and returns it to the source node. Figure 4 shows the route discovery phase of

DSR.

The second phase of the approach is route maintenance. When a node v_i forwards a data packet to node v_j it expects a confirmation from v_j . If v_i does not get any confirmation in a certain time interval, it will send a route error message RERR to the source node containing the link over which the forwarding has failed. Subsequently, the source node searches for an alternative in its routing table or initiates a new route discovery process.

Ad Hoc On-Demand Distance-Vector Protocol (AODV)

The Ad Hoc On-Demand Distance-Vector Protocol [11, 1, 12] (AODV) is a distance vector routing for mobile ad-hoc networks. AODV is an on-demand routing approach, i.e. there are no periodical exchanges of routing information. The protocol consists of two phases: i) route discovery, and ii) route maintenance.

A node wishing to communicate with another node first seeks for a route in its routing table. If it finds one the communication starts immediately, otherwise the node initiates a *route discovery* phase. The route discovery process consists of a route-request message (RREQ) which is broadcasted. If a node has a valid route to the destination, it replies to the route-request with a route-reply (RREP) message. Additionally, the replying node creates a so called *reverse route* entry in its routing table which contains the address of the source node, the number of hops to the source, and the next hop's address, i.e. the address of the node from which the message was received. A lifetime is associated with each reverse route entry, i.e. if the route entry is not used within the lifetime it will be removed.

The second phase of the protocol is called *route maintenance*. It is performed by the source node and can be subdivided into: i) source node moves: source node initiates a new route discovery process, ii) destination or an intermediate node moves: a route error message (RERR) is sent to the source node. Intermediate nodes receiving a RERR update their routing table by setting the distance of the destination to infinity. If the source node receives a RERR it will initiate

a new route discovery.

To prevent global broadcast messages AODV introduces a local connectivity management. This is done by periodical exchanges of so called HELLO messages which are small RREP packets containing a node's address and additional information.

3.2 Other ant based approaches

The application of ant algorithms for routing is not new. However, most approaches are either designed for fixed networks or assume special equipment to determine the location of a node, e.g. GPS.

Schoonderwoerd et al. propose in [13] the *Ant-Based Control (ABC) for network management*. The approach is destined for load balancing in circuit switched telecommunication networks and is based on regular transmission of ants from all nodes in the network to collect network information. Based on the collected information load balancing is performed.

Dorigo et al. introduce in [14] a routing approach for packet switched telecommunication networks which can be regarded as an extension of the previous approach. The authors distinguish between two different kinds of ants which are regularly transmitted, where one kind of the ants collect information in the network and the other kind refreshes routing information.

White suggests in [15, 16] another routing algorithm for circuit switched networks. The approach is based on three kinds of ants. The first class collects information, the second class allocates network resources based on the collected information and the third class gives allocated resources free after usage.

Câmara et al. introduce in [17] the *GPS/Ant-Like Routing Algorithm (GPSAL)*. The approach assumes that all mobile nodes have a GPS device and are able to determine their own positions. Additional information are also required, some of them are: the current and previous position of all other nodes, the node velocity and the stay time. The nodes must also be able to distinguish between mobile and static nodes.

These approaches are not appropriate for mobile multi-hop ad-hoc networks. The approaches for fixed networks are based on transmitting ants regularly

from all nodes. The generated network load is not suitable for ad-hoc networks. The other approaches assume special equipment for location determination, which is only valid for special networks.

4 THE ANT ROUTING ALGORITHM FOR MANETS

In this section we discuss the adaptation of the simple ant algorithm for mobile multi-hop ad-hoc networks and describe the Ant colony based Routing Algorithm (ARA). The routing algorithm is similar to many other routing approaches and consists of three phases.

4.1 Route Discovery Phase

New routes are created in the route discovery phase. The creation of new routes requires the use of a *forward ant* (FANT) and a *backward ant* (BANT). A FANT is an agent which establishes the pheromone track back to the source node. In analogous, a BANT establishes the pheromone track back to its origin, namely the destination node. The FANT is a small packet with a unique sequence number. Nodes are able to distinguish duplicate packets on the basis of the sequence number and the source address.

A node which receives a FANT for the first time creates a record in its routing table. An entry in the routing table is a triple (*destination address, next hop, pheromone value*). The node interprets the source address of the FANT as destination address, the address of the previous node as the next hop, and computes the pheromone value depending on the number of hops it took the FANT to reach the node. The node then relays the FANT to its neighbors. Duplicate FANTs are identified through the unique sequence number, and are removed. The destination node extracts the information of the FANT, creates a BANT and returns it to the source node. The BANT's task is similar to that of the FANT, i.e. to establish a track to this node. When the sender receives the BANT from the destination node, the path is established and data packets can be sent.

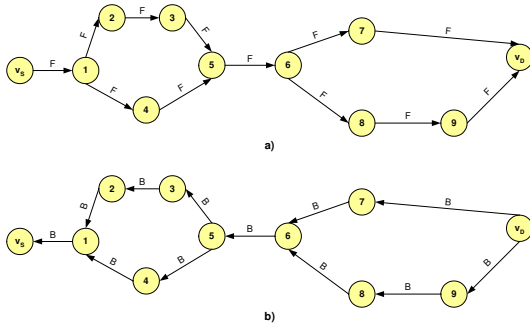


Figure 5: Route discovery phase in ARA. a) A forward ant (F) is sent from the sender (v_S) to the destination node (v_D). The forward ant passes through by other nodes which initialize their routing table and the pheromone values. b) The backward ant (B) has the same task as the forward ant. It is sent by the destination node (v_D) to the source node v_S .

Figure 5 demonstrates the route discovery phase of ARA. Figure 5 a) shows the establishment of the pheromone track back to the source node v_S . The forward ant only creates one pheromone track to the source node in node 6, but two tracks in node 5, via node 3 and node 4. Figure 5 b) depicts analogous situation for the backward ant. It only creates one pheromone track to the destination node v_D in node 5 and two tracks in node 6. Thus, multi-path routing is also supported by ARA.

4.2 Route Maintenance

The second phase of the routing algorithm is called route maintenance. This phase is responsible for the maintenance of the routes during the communication. ARA does not need any special packets for that purpose. Once the FANT and BANT have established the pheromone tracks for the source and destination nodes regular data packets are used to maintain the path. As in biological systems, established paths do not keep their initial pheromone values forever. When a node v_i relays a data packet to destination v_D to a neighbor node v_j , it increases the pheromone value of the entry (v_D, v_j, φ) by $\Delta\varphi$, i.e. this path to the destination is strengthened by the data packet. Like-

wise, the next hop v_j increases the pheromone value of the entry (v_S, v_i, φ) by $\Delta\varphi$, i.e. the backward path to the source node is also strengthened. The evaporation process of the real pheromone is modelled by decreasing the pheromone values according to equation 1.

The following is an example which is based on figure 5. Below you will find the routing tables of nodes 5 and 6.

Routing table of node 5		
Dest. Node	Next Hop	Pheromone
v_S	3	φ_1
v_S	4	φ_2
v_D	6	φ_3
\vdots	\vdots	\vdots

Routing table of node 6		
Dest. Node	Next Hop	Pheromone
v_S	5	φ_4
v_D	7	φ_5
v_D	8	φ_6
\vdots	\vdots	\vdots

Now we consider the routing tables of both nodes after node 5 has forwarded a data packet to node 6. Only the entry for the destination node v_D has changed in the routing table of node 5. In the routing table of node 6 the changes are analogous, only the pheromone value for the source node v_S has changed. This is done in the same way as for the destination in node 5. In our simulations $\Delta\varphi$ has a value of 0.1.

Routing table of node 5		
Dest. Node	Next Hop	Pheromone
v_S	3	φ_1
v_S	4	φ_2
v_D	6	$\varphi_3 + \Delta\varphi$
\vdots	\vdots	\vdots

Routing table of node 6		
Dest. Node	Next Hop	Pheromone
v_S	5	$\varphi_4 + \Delta\varphi$
v_D	7	φ_5
v_D	8	φ_6
\vdots	\vdots	\vdots

The pheromone values are decreased in regular time intervals of $\tau = 1$ seconds. The decreasing is done multiplicative with a factor of $(1 - q) = 0.1$ (see equation 1). The routing tables of the both node will look after a increasing process as follows.

Routing table of node 5 (after τ seconds)		
Dest. Node	Next Hop	Pheromone
v_S	3	φ_1
v_S	4	$\varphi_2 \cdot (1 - q)$
v_D	6	$(\varphi_3 + \Delta\varphi) \cdot (1 - q)$
\vdots	\vdots	\vdots
\vdots	\vdots	\vdots

Routing table of node 6 (after τ seconds)		
Dest. Node	Next Hop	Pheromone
v_S	5	$(\varphi_4 + \Delta\varphi) \cdot (1 - q)$
v_D	7	$\varphi_5 \cdot (1 - q)$
v_D	8	φ_6
\vdots	\vdots	\vdots
\vdots	\vdots	\vdots

The above method for route maintenance could lead to undesired loops. ARA prevents loops by a very simple method which is also used during the route discovery phase. Nodes can recognize duplicate receipts of data packets, based on the source address and the sequence number. If a node receives a duplicate packet, it will set the `flag_Duplicate` flag and send the packet back to the previous node. The previous node deactivates the link to this node, thus data packets cannot be sent to this direction any more.

4.3 Route Failure Handling

The third and last phase of ARA handles routing failures which are especially caused by node mobility and are therefore very common in mobile ad-hoc networks. The current implementation of ARA assumes IEEE 802.11 on the MAC layer. This enables ARA to recognize a route failure through a missing acknowledgement on the MAC layer. If a node receives a packet with set `flag_RF`, it first deactivates this link by setting the pheromone value to 0. Subsequently, the node searches for an alternative link in its routing table. If there is another route to the destination it will send the packet via this path. If there are several entries in the routing table, the node will not try all. We figured out, that only one additional try makes sense. Otherwise, the node informs source node, which has to initiate a new route discovery process.

4.4 Improvements to the first version

In the current version of the algorithm, we introduced some new features which influence the performance

of the routing algorithm positively in comparison to the first version [6]. We will refer to the basic idea and the first implementation as ANT and to the current version as ARA to distinguish between them.

- **Probabilistic vs. dynamic routing:** In ANT the selection of the next hop for a data packet was always decided by the amount of pheromone values, i.e. a node v_i selects a neighbor v_j with probability $p_{i,j}$ as follows

$$p_{i,j} = \begin{cases} 1 & , \text{ if } \varphi_{i,j} = \max_{k \in N_i} \{ \varphi_{i,k} \} \\ 0 & , \text{ else} \end{cases}$$

ARA is extended to use probabilistic routing, i.e. a node v_i selects a neighbor v_j with probability $p_{i,j}$. N_i is the set of one step neighbors of v_i .

$$p_{i,j} = \frac{\varphi_{i,j}}{\sum_{k \in N_i} \varphi_{i,k}}, \quad j \in N_i, \quad \sum_{k \in N_i} p_{i,k} = 1$$

In section 5, ARA with probabilistic routing is denoted by ARA_{stat} . The main advantage of using probabilistic route selection is that the load is distributed over the existing paths to the destination.

- **Continual decreasing of pheromone values:** In ANT the decreasing of the pheromone values were simulated in intervals of a second. We have figured out that this forges the alteration of two paths which differ only within an interval. To better adapt the decreasing of the pheromone values to the natural behavior, the pheromone values are now decreased continually.
- **Prioritized packets:** In the ant-routing-algorithm the FANT and the BANT packets have particular jobs which is vital for the operation of the network. In ANT these packets were handled like ordinary data packets. Now FANTs and BANTs are prioritized and nodes handle them with special care. In addition, packets with the set loop or error flag are also prioritized. This guarantees that important information are delivered as fast as possible in the network.

- **Flooding of BANTs:** In ANT the BANT were sent by unicast from the target to the source node. Hence, only one path between the target and source node was built. This restricted the exploitation of the multi path property of ARA. In ARA, the BANT is also flooded from the target to the source node similar to the FANT. This leads to a better exploitation of the multi-path capability of ARA.
- **Send buffer and mini-FANTs:** Another problem in ANT was caused by some unnecessary broadcast messages in special network constellations. There were two reasons for this problem. The first reason was that ARA used the first packet of a connection as the FANT without considering the packet length. The second reason was, if during the path discovery phase several packets for the same connection arrived on the network layer, more than one FANT was created and sent out.

ARA has a send buffer which holds the packets during the path discovery phase. Additionally, ARA does not use the first packet of a connection as a FANT, instead, similar to the BANT a minimal packet is created for this purpose. This procedure leads to a faster transmission of the FANT throughout the network, hence accelerating the path discovery.

- **MAC-Tap:** The MAC-Tap extracts information from packets from the neighborhood. The collected information is passed to the network layer for further routing decisions. The current MAC-Tap is adapted to the MAC of IEEE 802.11, i.e. CSMA/CA, but it might work also with other MAC approaches.

4.5 Properties of ARA

According to [18] a routing algorithm for mobile ad-hoc networks should meet the following requirements, which ARA does:

- **Distributed operation:** In ARA, each node owns a set of pheromone counters $\varphi_{i,j}$ in its routing table for a link between node v_i and v_j . Each

node controls the pheromone counter independently when ants visit the node on route search for, or when the node detects a link failure.

- **Loop-free:** The use of unique sequence numbers of route finding packets avoids loops.
- **Demand-based operation:** Routes are established by manipulating the pheromone counters $\varphi_{i,j}$. Over time, the amount of pheromone decreases to a minimum value when ants do not visit this node. A route finding process is only run upon demand by a sender.
- **Sleep period operation:** Nodes are able to sleep when the amount of pheromone in their routing table has reached a lower threshold. Other nodes will then not consider this node, unless packets are destined to it.

Additionally, ARA has the following properties:

- **Locality:** The routing table and the statistic information block of a node are local, and they are not transmitted to any other node.
- **Multi-path:** Each node can have several paths to a certain destination. The choice of a certain route depends on the environment, e.g. on the link quality to the relay node.
- **Sleep mode:** In sleep mode a node only processes packets which are destined to it. This saves energy and power.

5 SIMULATION RESULTS

5.1 Simulation Environment

We have implemented ARA in ns-2 [19]. For our results we assumed 50 mobile nodes communicating via IEEE 802.11. The nodes move inside a simulation area of 1500m×300m. The simulation time is 900 seconds.

The nodes move with a maximal velocity of 10 m/s and according to the random waypoint mobility model [20]. In this model, a node randomly chooses a point in the simulation area and a speed for the next

move which is uniformly distributed between 0 and the maximal velocity. Subsequently, the node drives to the selected point at constant speed. After arriving at the end point the node remains there for a certain time. Subsequently, the node repeats the operation by selecting a new end point and a new speed. We performed simulations with 7 different pause times of 0, 30, 60, 120, 300, 600 and 900 seconds. When the pause time is 0 seconds, the nodes move constantly. In contrast, when the pause time is 900 seconds the nodes do not move at all.

5.2 Comparison of ARA with the first version

In this section we will compare the performance of the current version of ARA with the first version which is denoted as ANT in the graphs.

Figure 7 depicts the delivery rate of the first version (ANT) and the current version of ARA with its extensions. There are three graphs depicted for ARA, these are: ARA_{stat} denotes ARA with statistical path selection, $ARA_{stat,sb}$ depicts ARA with send buffer and $ARA_{stat,sb,tap}$ depicts the results with the additional MAC-Tap.

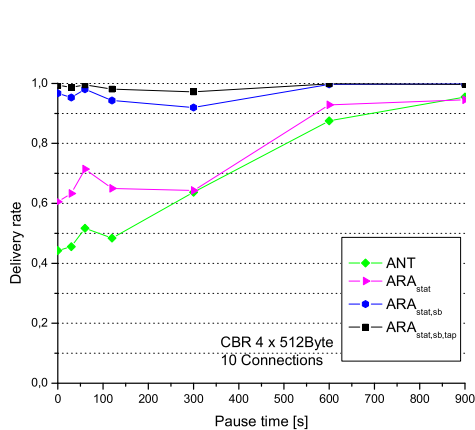


Figure 6: Comparison of ANT and ARA by the fraction of successfully delivered packets as a function of pause time. Simulations with 10 CBR connections.

The first extension, i.e. ARA_{stat} , shows a performance similar to ANT in the less mobile scenarios,

but the performance is better in highly mobile scenarios. The results of ARA with send buffer is very different. In nearly all cases the performance is between 90% - 100%. The reason for this performance gain is based on preventing additional broadcasts and the accelerated path discovery through minimal FANTs. The version with all features on, including the MAC-Tap, shows a slightly better performance. The source node might be able to tune the routing table by collecting information from the neighborhood. From now on, the version with all features on will be denoted as ARA.

Figure 7 shows the path optimality for ANT and ARA. The path optimality is given by the number of the hop difference compared to the shortest path between the source and destination node. A difference of one means that the used path was one hop longer than the shortest path. ARA has learned much in terms of path optimality, it transports nearly 85% of the data packets over an optimal path. In comparison, ANT transports only 28% of the data packets over an optimal long path, and the most packets are transported over a path with a one hop deviation.

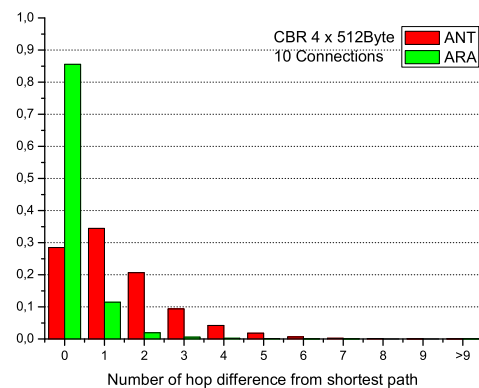


Figure 7: Comparison of ANT and ARA by path optimality. Simulations with 10 CBR connections.

The performance improvement is based on the improved maintenance of the pheromone values which leads to stronger amplification of young paths and the exploitation of the multi-path capability of ARA, i.e.

the load is balanced towards the destination. Additionally, the use of the send buffer together with minimal packets as FANTs leads to a fast path discovery, and prevents the sending of successive FANTs for the same connection.

5.3 Comparison of ARA with DSR and AODV

To get a better idea of the performance of ARA we compared it with DSR and AODV, which were briefly presented in section 3. We present results obtained by simulations of 10 parallel connections with constant bit rate (CBR) traffic over UDP. The parameters are similar to those in [20].

We will first discuss the robustness of the routing protocols. Figure 8 shows the delivery rate, i.e. the amount of packets a certain routing protocol was able to deliver properly. In the cases of low mobility the performance is very similar for the three routing protocols. This changes with increasing mobility. In the cases of high mobility ARA and AODV show better performance than DSR.

Figure 9 shows the path optimality of the three routing protocols. The graph confirms the results shown in figure 8. It is obvious that all routing protocols transport most data packets over shortest paths, but the fraction differs: 85% for ARA, 75% for DSR, and 57% for AODV. It is interesting to note that AODV, although transporting only 57% of data packets over an optimal path, has a higher delivery rate than DSR.

We will now consider the overhead of the routing protocols based on the ratio of routing bytes to data bytes. Figure 10 depicts the routing overhead for the three routing protocols. In high mobility situations, AODV and ARA produce the smallest overhead. In the cases of high mobility ARA shows the best performance followed by AODV. In cases of low mobility the routing overhead of AODV is less than the overhead of ARA. In all cases the the DSR shows the highest routing overhead. The reason is that DSR needs to put the whole path information into each data packet, even if no mobility occurs. Overhead in AODV is generated by the exchange of routing

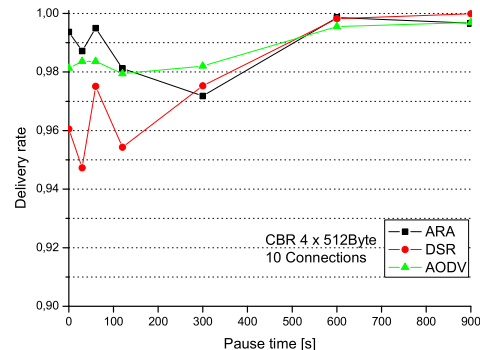


Figure 8: Comparison of DSR, AODV and ARA by the fraction of successfully delivered packets as a function of pause time. Simulations with 10 CBR connections.

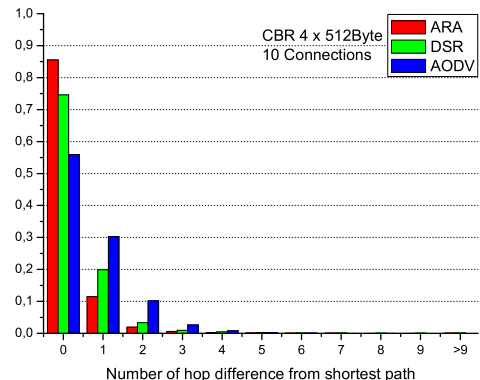


Figure 9: Comparison of DSR, AODV and ARA protocols the path optimality, i.e. number of hop difference from shortest path. Simulations with 10 CBR connections.

tables which rarely occurs in cases with low mobility. In ARA the overhead is generated during the path discovery phase and the route failure handling by the FANT and the BANT. The path discovery is performed for each new connection. With less mobility the number of route failure also decreases and therefore generates low overhead.

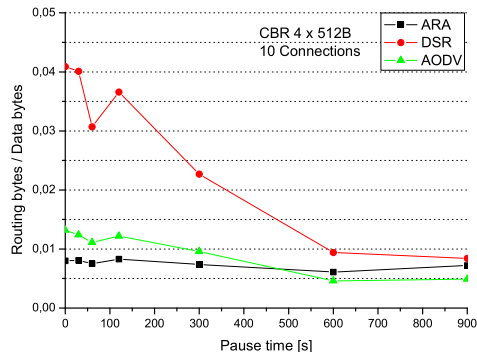


Figure 10: Comparison of DSR, AODV and ARA by the fraction of successfully send bits and the needed routing bits as a function of pause time. Simulations with 10 CBR connections.

6 CONCLUSIONS AND FUTURE WORK

Mobile multi-hop ad-hoc networks are flexible networks which do not require a pre-installed infrastructure. With the upcoming wireless transmission technologies and highly sophisticated devices their application will increase. However, routing is a major challenge in mobile multi-hop ad-hoc networks which is aggravated by node mobility.

In this paper we presented a new on-demand routing approach (ARA) for mobile multi-hop ad-hoc networks which is based on ant algorithms. The first version of the algorithm performed poorly in highly mobile scenarios. In this paper we introduced some new features which improve the performance of the algorithm in all scenarios especially in those with high mobility.

Despite the improvements the need to flood the network is a big disadvantage and not desired in mobile ad-hoc networks. A possibility to diminish this problem might be to use a flooding technique with less overhead. We will concentrate our further studies on the reduction of the overhead required and on the maintenance of the pheromone values.

References

- [1] Charles Perkins, editor. *Ad Hoc Networking*. Addison-Wesley, 2001. ISBN.
- [2] Chai-Keong Toh. *Ad hoc mobile wireless networks: protocols and systems*. Prentice Hall, 2002. ISBN:.
- [3] R. Droms. Dynamic host configuration protocol, RFC 2131. <http://www.ietf.org/rfc/rfc2131.txt>, March 1997.
- [4] S. Thomson and T. Narten. Ipv6 stateless address autoconfiguration, rfc 2462. <http://www.ietf.org/rfc/rfc2462.txt>, December 1998.
- [5] Mesut Günes and Jörg Reibel. An IP Address Configuration Algorithm for Zeroconf. Mobile Multi-hop Ad-Hoc Networks. In *Proceedings of the International Workshop on Broadband Wireless Ad-Hoc Networks and Services*, Sophia Antipolis, France, September 2002. ETSI.
- [6] Mesut Günes, Udo Sorges, and Imed Bouazizi. ARA - The Ant-Colony Based Routing Algorithm for MANETs. In Stephan Olariu, editor, *Proceedings of the 2002 ICPP Workshop on Ad Hoc Networks (IWAHN 2002)*, pages 79–85. IEEE Computer Society Press, August 2002.
- [7] Marco Dorigo and Gianni Di Caro. The ant colony optimization meta-heuristic. In David Corne, Marco Dorigo, and Fred Glover, editors, *New Ideas in Optimization*, pages 11–32. McGraw-Hill, London, 1999.
- [8] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm intelligence: from natural to artificial intelligence*. Oxford University Press, 1999. ISBN.
- [9] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. In T. Imielinski and H. Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

- [10] D. B. Johnson, D. A. Maltz, Y.-C. Hu, and J. G. Jetcheva. The dynamic source routing protocol for mobile ad hoc networks. IETF Internet draft, draft-ietf-manet-dsr-04.txt, November 2000.
- [11] Charles E. Perkins and Elizabeth M. Royer. Ad hoc on-demand distance vector routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, New Orleans, LA, February 1999. <http://moment.cs.ucsb.edu/AODV/aodv.html>.
- [12] C. E. Perkins, E. M. Royer, and S. R. Das. Ad hoc on-demand distance vector (AODV) routing. IETF Internet draft, draft-ietf-manet-aodv-11.txt, June 2002.
- [13] Ruud Schoonderwoerd, Owen E. Holland, Janet L. Bruten, and Leon J. M. Rothkrantz. Ant-based load balancing in telecommunications networks. *Adaptive Behaviour*, 2:169–207, 1996.
- [14] Marco Dorigo, Vittorio Maniezzo, and Alberto Colomi. Ant system: Optimization by a colony of cooperating agents. *IEEE Trans. on Systems, Man, and Cybernetics—Part B*, 26(1):29–41, 1996.
- [15] Tony White. Routing with swarm intelligence. Technical Report SCE-97-15, 1997. <http://citeseer.nj.nec.com/white97routing.html>.
- [16] Tony White. Swarm intelligence and problem solving in telecommunications. *Canadian Artificial Intelligence Magazine*, Spring 1997.
- [17] Daniel Câmara and Antonio A.F. Loureiro. A novel routing algorithm for ad hoc networks. In *Proceedings of the 33rd Hawaii International Conference on System Sciences*, Maui,, January 2000.
- [18] Joseph P. Macker and M. Scott Corson. Mobile ad hoc networking and the IETF. *Mobile Computing and Communications Review*, 2(1):9–14, 1998.
- [19] Kevin Fall and Kannan Varadhan. *The ns Manual*, Nov 2000.
- [20] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A performance comparison of multihop wireless ad hoc network routing protocols. *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'98)*, pages 85–97, 1998.