A Mobile Ad-hoc Network has limited and scarce resources and thus routing protocols in such environments must be kept as simple as possible.

The Simple Ant Routing Algorithm (SARA) offers a low overhead solution, by optimizing the routing process. Three complementary strategies were used in our approach: during the route discovery we have used a new broadcast mechanism, called the Controlled Neighbor Broadcast (CNB), in which each node broadcasts a control message (FANT) to its neighbors, but only one of them broadcast this message again. During the route maintenance phase, we further reduce the overhead, by only using data packets to refresh the paths of active sessions. Finally, the route repair phase is also enhanced, by using a deep search procedure as a way of restricting the number of nodes used to recover a route. Thus, instead of discovering a new path from the source to the destination, we start by trying the discovery of a new path between the two end-nodes of the broken link. A broadest search is only executed when the deeper one fails to succeed.

We simulated our proposal and we tuned it to the optimal performance. We also compared it with the classical approach of AODV and other biological routing approaches. The results achieved show that SARA offers the smallest overhead of all the protocols under evaluation and presents an overhead reduction of almost 25% of the value achieved by the other proposals. SARA also presents the best goodput, specially for TCP traffic, but it needs more time to discover the routes.
approaches are needed to overcome the difficulties. Swarm Intelligence-based Routing with Opportunistic Routing represent another set of algorithms based on biological models, inspired by highly dynamic environments, which are particularly adequate for MANETs.

By merging the simplicity of the Swarm Intelligence-based Routing, with the efficiency of the dissemination strategy of the Opportunistic Routing, the authors designed the Simple Ant Routing Algorithm (SARA), which is a low overhead MANET’s routing protocol. In this paper we present a complete study of SARA and we show that the network performance may be optimized when the routing algorithm parameters are properly selected. Comparison with the results of other routing protocols also shows the advantage of our approach.

The remainder of the paper is organized as follows. In Section 2 the work related to routing algorithms and broadcast schemes is presented. Section 3 describes the SARA architecture, comprising the route discovery, maintenance and repair mechanisms, as well as the parameters to be taken into consideration in each phase. This architecture was evaluated through simulation and some relevant results are shown in Section 4. Finally, Section 5 presents the conclusions reached and the planned future work.

2. Related work

2.1. Routing algorithms

Table-driven represents routing algorithms that are based on the classical wired approach. Using a purely pro-active strategy, they calculate and maintain routes to all possible destinations, which are stored and kept up-to-date at each node. Examples of such approach are used by the Dynamic Destination Sequenced Distance Vector Routing (DSDV) [4] and the Optimized Link State Routing (OLSR) [5]. These types of algorithms use a burden of control information to keep the network topology up-to-date. However, as the node mobility increases, keeping track of topology variations becomes difficult to achieve. This leads to a worse network performance that makes them particularly less adequate for use in MANETs.

Demand-driven algorithms represent another sort of solution where the source node initiates the routing discovery when there is no path available to the destination node. In such case, the route is maintained until the data flow ends its activity or one of its nodes becomes inaccessible. Examples of such kind of algorithms are the Ad-hoc On Demand Distance Vector Routing (AODV) [6], which is based on DSDV and the Dynamic Source Routing (DSR) [7–9], which uses the classical source routing approach. Both of them lack an efficient support of MANETs, as they require the use of a significant amount of control information during the route discovery process.

In the Hybrid strategy, the network is divided into clusters and different routing protocols may be used for inter and intra-cluster routing: the goal is to find out an optimal solution by combining both types of strategies. Examples of this kind of routing are the Zone Routing Protocol (ZRP) [10,11], the Independent Zone Routing (IZR) [12] which is an enhancement of the ZRP. In spite of the advantage of this flexible approach, small or highly dynamic networks may not be able to take advantage of hybrid strategies due to the overhead associated with cluster creation and maintenance.

By taking a quite different approach, insect societies have become a source of inspiration for routing since Dorigo et al. published a seminal paper, “Distributed Optimization by Ant Colonies”, in 1992 [13], leading to a new group of proposals named Swarm Intelligence-based Routing [14]. As it can be easily observed, real ants can converge on the shortest path that connects their nest to a source of food. This behavior is caused by a chemical substance, the pheromone: while moving, the ants deposit the pheromones and tend to follow the paths with the highest intensity of pheromones. The paths that attract more ants will experience an increasing level of pheromones, until the majority of the ants converge on the shortest path. This indirect communication process used by the ants, which modify the environment and react to these modifications, is known as stigmergy [15]. By modeling the ants’ behavior, routing agents and data packets can act as ants leaving a pheromone trail as they pass through the path between the source and the destination. The path is marked without more control packets being introduced into the network, resulting in lower overhead. Different routing protocols like the Ant Routing Algorithm (ARA) [16] based on Ant Colony Optimization [17], or the AntHocNet [18,19] are based on such kind of models. The Cross Entropy Ant System (CEAS) [20,21] is also a multi-agent swarm intelligence system based on ant’ behavior, which falls in the same routing class of ACO system. CEAS is based on cross entropy proposed by Rubinstein et al. [22]. In spite of the advantage of these algorithms, the overhead associated with the path discovery is still high, mainly in highly dynamic MANETs.

The Opportunistic Routing is a recent routing concept which has been applied to wireless networks and which explores the functionalities of broadcast transmission to forward data packets, instead of using a path discovery mechanism: if one node misses a transmission to the next hop, there is at least another node in the neighborhood which can send that missing packet to the destination [23]. When using such approach, the node’s memory will be exhausted, as un delivered data packets need to be stored. Using the probability of success in forwarding information through a specific link towards the destination, this limitation can be minimized. However, as topology knowledge is required, the overhead of this process increases significantly when nodes’ mobility increases.

2.2. Broadcast schemes

On wireless Ad-hoc networks, due to the node’ mobility and the frequent change of the network topology, the use of broadcast schemes is the fastest method to reach all the network nodes. The broadcast is a simple mechanism for delivering control or data packets by flooding the network [24]. However, the flooding procedures of broadcast will increase the overhead and the congestion and will
cause extra power consumption. These factors will reduce the network’s data transport capability.

Several broadcast solutions have been presented to improve the broadcast scheme efficiency and reduce the network overhead. The Smart Broadcast (SB) scheme presented in [25], decreases the overhead by reducing the number of the network nodes which are allowed to re-broadcast the packets. This is done by estimating the distance to the source node. The nodes on the radio transmission border will re-broadcast the packet before the ones near the source. The estimation procedure uses the free space propagation model based on the carrier wavelength and the antenna gain. Every time these two parameters change, the SB should be tuned.

Another solution to improve the broadcast scheme is the Urban Multi-hop Broadcast (UMB) [26]. It reduces the network congestion through the introduction of new MAC control packets, called Request-to-Broadcast (RTB) and Clear-to-Broadcast (CTB). These packets contain the source’s geographical information. Upon the reception of the RTB packet by the neighbor nodes, they must compute their distance to the source node and transmit a jamming signal, called black-burst. The signal duration is proportional to their distance to the source node. Once a node has finished its black-burst transmission, it checks the channel status. If the transmission channel is clear, the node knows it is on the transmission border and sends the CTB packet.

The Geographic Random Forwarding (GeRaF) [27] is a position-based routing protocol with similar working procedures to UMB’s. It uses the Request-to-Send (RTS) and Clear-to-send (CTS) message to announce the broadcast request. Upon receiving the RTS message, all neighbor nodes must reply with a CTS message. In the case of collision, the source node issues a Collision message and all the nodes use a probabilistic rule to respond to the source node. In Elana Fasolo et al. [28], a broadcast scheme which is similar to the UMB is also presented. However, it uses the RTS and the CTS messages to identify the node’s position related to the source.

The routing model proposed by the authors also uses broadcast schemes to transmit the route discovery packets. In order to avoid the network flood, a simple broadcast procedure is used, called as Controlled Neighbor Broadcast (CNB), developed by the authors. Unlike the broadcast mechanism described above, our solution does not use any sort of extra information, such as the transmission radio of SB, the new control packets of UMB or the geographical location of the GeRaF. We used standard broadcast mechanism, as we do want to be able to use existing wireless interface cards, without any modification. To control the flooding network level, our CNB uses a probabilistic mechanism to determine which neighbor node will rebroadcast the control message.

### 3.1. Route discovery

In the traditional ACO framework, a source node starts a route discovery process by sending a special control packet, the Forward ANT (FANT) [16,18], which is replicated by all network nodes until it reaches the destination neighborhood. Upon the reception of the first FANT, the destination node will send another special control packet back, the Backward ANT (BANT), through the shortest known path. Should this packet arrive at the source, the path is established and the data flow may start its activity. To implement such mechanism, the AntHocNet uses a source-routing approach and requires a two-way route discovery procedure (from source node to destination node and vice-versa), leading to a significant amount of control information when the path is long.

In this model, the FANT is replicated in every node and the network is flooded with control information, which deteriorates its performance, as shown by the authors in [29]. SARA introduces a more efficient mechanism to disseminate the FANTS: the Controlled Neighbor Broadcast (CNB), in which each node broadcasts the FANT to all of its neighbors, who process it, but only one of them broadcasts the FANT again to its own neighborhood.

One key aspect in this process is the selection of the node responsible for “re-broadcasting” the FANT. The policy used is to select different nodes each time a FANT is generated using a probabilistic approach described next. With this, it is possible to maximize the number of discovered paths.

So, let us consider a network (see Fig. 1) represented as a direct weighted graph $G = (V, E)$, where $V$ denotes the set of vertices and $E$ denotes the set of edges with weight function $w : E \rightarrow R$, also considering a source node $s \in V$, a destination node $d \in V$ and a generic node $u \in V$. $\text{Adj}[u]$ is the list of adjacencies of node $u$ containing all the vertices $J_0, \ldots, J_3$ on node $u$ neighborhood. The probability, $p_{(u,j,d)}$, to choose a given node $j$, as the next hop to forward the FANT towards destination $d$ is given by following equation:

$$\forall j_i \in \text{Adj}[u], \exists p_{(u,j_i,d)} :$$

$$p_{(u,j,d)} = \frac{C_{(u,j,d)}}{\sum_{k=0}^{M} C_{(u,k,d)}} \land C_{(u,j,d)} = \frac{1}{1 + n},$$

where $C_{(u,j,d)}$ is the cost of each link $u \rightarrow j_i$ related to the number of times ($n$) the link was previously selected and $M$ is the number of adjacencies of node $u$. In Fig. 1, the number of $u$ adjacencies is 4.

To explain how CNB works, we must observe Fig. 2 and see how a route is discovered. In particular, we are interested in presenting how the FANT message is propagated through the network, and so, our concern is to analyze

---

3. SARA architecture

The next section details the SARA architecture, concerning the four phases of routing in MANETs: Route Discovery, Route Maintenance, Route Selection and Route Repair.

---

Please cite this article in press as: F. Correia, T. Vazão, Simple ant routing algorithm strategies for a (Multipurpose) MANET model, Ad Hoc Netw. (2010), doi:10.1016/j.adhoc.2010.03.003

---

**Fig. 1.** Network diagram.
the procedures used by a node responsible to process the control messages.

When node $s$ requires a path to $d$, it starts to send FANT packets to the network. $s$ must select the next hop to forward the FANT and then broadcast it. $s$ has two adjacencies: node $A$ and node $B$. Because this is the first time, the cost of $C_{s,A,D} = C_{s,B,D} = 0.5$ in accordance with Eq. (1) using $n = 1$, and thus, node $s$ will choose node $A$ with $P_{s,A,D} = 0.5$. Upon FANT transmission, two timers are triggered: the route discovery confirmation timer $T_0$ and the FANT confirmation timer $T_1$.

The $T_0$ is responsible for the FANT generation. Each time the timer ends and the source node does not have an indication that the route to the destination is established, a new FANT is created. Only the source node has the ability to start this timer. The $T_1$ is initiated by all network nodes which are responsible for forwarding the FANT, including the source node. $T_1$ is cancelled upon the reception of an acknowledgment packet ($C_{FANT,n}$) sent by the next forwarding node. If the timer expires before this packet is received, a copy of the FANT is transmitted.

In Fig. 2, $T_0$ expires, and so, node $s$ must generate a new FANT (FANT_2) and broadcast it to the network. Because it is the second attempt to transmit a FANT, the next hop is selected in accordance with the cost of the link as shown in Table 1.

With these values, the probability of a link being selected decreases as the selection times increase. So node $B$ can be selected with a higher probability ($P_{s,B,D} = 0.6$) than node $A$ ($P_{s,A,D} = 0.4$). During the FANT broadcast in the network, all nodes use this algorithm to select the next FANT transmission hop. As shown in Fig. 2, all network nodes that have the responsibility to rebroadcast the FANT must set a timer $T_1$. If the node does not receive the FANT forwarding confirmation ($C_{FANT,n}$) until this timer is triggered, another copy of the FANT is sent to the network through a link selected with a probability calculated with Eq. (1).

To improve the efficiency of this scheme, instead of generating a new control confirmation packet, SARA uses a virtual FANT called $C_{FANT,n}$ (represented by the dotted line in Fig. 2). This message is the FANT_n being rebroadcast by another network node. This is an indication the FANT_n is still active and traveling through the network. The timer $T_1$ is cancelled upon receiving the $C_{FANT,n}$. The FANT_n confirmation can also take place through the reception of a BANT message. This situation can be seen in Fig. 2, where node $u$ does not receive the $C_{FANT,1}$, instead receives BANT_1 which will act as a confirmation message.

Due the FANT broadcast transmission mode, upon receiving the FANT message, any network node with a valid destination route information must generate a BANT and transmit it to the source node through the shortest path. The shortest path is given by minimum number of hops from the current node to the source node. The FANT message continues traveling in the network, in search of another routes, until it reaches the destination node or the node responsible to forward the FANT has a valid route to the destination node. In Fig. 2, node $j_0$ is responsible to forward the FANT_1(2) and has a route to destination node $d$. The FANT is removed and a BANT is sent backwards to the source node $s$. Once again, FANT_2 sent by node $B$ when is processed by node $j_2$, the FANT is removed due the presence of a valid route to destination. Although node $j_2$ has two shortest paths towards node $s$ (one through node $A$ and another one through node $B$) is using the backward path set by FANT_1 through node $A$ to reach the source node $s$, an explicit confirmation FANT is sent to node $B$ with the aim to stop the $T_1$ timer (this message is represented by a solid line Fig. 2).

As the FANT travels through the network, all nodes that received the FANT (in broadcast) have the responsibility to update the source node route entry with the minimum hop-value found. At this phase, only the shortest routes must be memorize, the longer routes are erased. This procedure is used to form the network topology and avoid possible route loops. The hop count is given by the Hop-Count field in the FANT header, which is incremented by one unit every time the FANT is retransmitted.

The duration of the timers is critical for SARA’s performance. The time must be adjusted. The nodes must have enough time to compute the FANT and to transmit it successfully to the next node. If the time value is too small, the overhead increases due to extra FANT messages traveling in the network. But if it is too high, SARA will take more time to determine that FANT messages were lost and the route establishment time will increase.

### 3.2. Route maintenance

The ACO framework comprises a route maintenance process, responsible for keeping the information about the active routes updated, by adjusting the pheromone level. This level is an indicator of the activity and the quality
of a link: the most used links must experience the highest pheromone levels, while the unused ones will have the lowest values. Such behavior is achieved with the use of two complementary mechanisms:

Increase pheromone intensity – every packet (data or control) that crosses a link increases the pheromone intensity of the backward link by a given amount \( x \) (for instance, a packet transmitted in link \((i,j)\) increases the pheromone intensity of link \((j,i)\) when it reaches node \(j\)).

Decrease pheromone intensity – as time goes by the pheromone level decreases automatically by \( \gamma \) value.

A path with a higher pheromone level is a path with the ability to attract more traffic. However, as the available bandwidth is limited, the intensive use of a route leads to congestion and to an increasing number of collisions. Under these circumstances, the throughput will tend to decrease and as consequence, the pheromone refreshment rate also will decrease. Due the pheromone evaporation, if the pheromone intensity decrease on a link, this could indicate link’ problems. As already studied by the authors in [30], the analysis of the pheromone variation gives significative information about the network state and might be used in early congestion detection.

In the classical ACO framework, no additional traffic is created to maintain a route. However, to trace a route, the pheromone level must be reinforced periodically in both directions \((s \rightarrow d \text{ and } d \rightarrow s)\). As most of the traffic generated by client-server applications is strongly biased towards one direction and UDP traffic is purely asymmetric, using this strategy to maintain the path will lead to an inaccurate view of the network. AntHocNet solves this problem by producing additional control traffic to maintain the routes active. In SARA this control traffic is only used in active data sessions, when is detected a strong asymmetric traffic, in which the pheromone intensity present in one direction could reach ‘0’. In this situation, SARA uses a special FANT message, named as super FANT. The super FANT is a FANT generated by the end node (source or destination) where the asymmetric traffic was detected. It has a pheromone reenforce equivalent capability of \( n \) standard FANT. The super FANT is generated with a lower frequency than the arriving packet rate at the destination node.

Let us consider a generic packet \( pkt \) (data or control), crossing a link \( u \rightarrow j \), at time \( t \) and a constant value \( x \). Where \( t \) is the time related with most recent event: a packet cross the link at time \( T_i \) or a pheromone decrease event at \( \tau_i \). The pheromone intensity \( ph \) increases as described in:

\[
\text{Increase}:
\begin{align*}
\forall pkt(\tau_i), & \quad ph_{(u,j,i)} = ph_{(u,j,i)} + x,\\
& \text{where:} \\
& t = T_{i-1}, \quad \text{if} \quad T_{i-1} > \tau_{i-1} \\
& t = \tau_{i-1}, \quad \text{if} \quad T_{i-1} < \tau_{i-1}.
\end{align*}
\]

The pheromone level has a lifetime, and, when its value reaches zero, it indicates that the paths associated with those links are not used. Thus, in every \( \tau_i \), the pheromone level is adjusted as described in:

\[
\text{Decrease}:
\begin{align*}
\forall pkt(\tau_i), & \quad ph_{(u,j,i)} = \begin{cases} 
  ph_{(u,j,i)} - \gamma, & \text{if } ph_{(u,j,i)} > \gamma, \\
  0, & \text{if } ph_{(u,j,i)} \leq \gamma.
\end{cases}
\end{align*}
\]

Fig. 3 presents the link pheromone evaluation for the increase and decrease phases. The increase period is set by \( T_i \) (session data rate), and the decrease period is constant by \( \tau \) time.

The values of \( x \) and \( \gamma \) are fundamental issues in the dimensioning of SARA, as they dictate the number and the characteristics of available paths. If \( x \) is too small and \( \gamma \) is too big, the pheromone intensity will increase very slowly, and the paths may become unstable. On the other hand, a high value of \( x \) and a reduced value of \( \gamma \) create stable paths, which hardly react to topological variations. The tuning of these two values is carried out off-line in simulated scenarios. The optimum value found is the one that can maintain the route available through the link pheromone activity. It also can be set for specific scenarios. Intermittent connections are not affected and the pheromones shall remain active. The \( \tau \) value is also critical, but this value can be adjusted dynamically to the network state. When the pheromone level associated with a link reaches zero, it means the route is not used but could be valid in the network, so the route is not deleted from the routing table. To control the routing table size, SARA uses the Least Recently Used (LRU) algorithm to sort the routes and those with zero pheromones are candidates for deletion.

### 3.3. Route selection

The route selection is a probabilistic procedure used to choose the next hop to forward traffic to the destination, which is given by

\[
\forall j_i \in Adj[u], \exists p_{(u,j,d)} : \Phi_{(u,j,d)} = \frac{\phi_{(u,j,d)}}{\sum_{k=0}^{M} \phi_{(u,j,d)}}.
\]

The probability \( p_{(u,j,d)} \) concerning the selection of the link to node \( j \), towards destination node \( d \) by node \( u \) is related to the link cost \( \Phi_{(u,j,d)} \), where \( M \) is the number of adjacencies of node \( u \).

In the classical ACO framework these probabilities are dictated only by the pheromone level. Thus, in the case of different paths with similar probabilities, no distinction
can be made among them, even if they have different lengths.

We solved this problem, as our approach uses two different metrics: the number of hops from node \( ji \) to destination node \( d(\text{nh}_{ji,d}) \) and the pheromone level present in the link between node \( u \) and node \( ji \), towards destination node \( d(\text{ph}_{ji,d}) \). SARA weighted these two metrics differently, by using a power factor called \( F \) (convergence factor) and an exponential function to enhance the number of hops, described as follows:

\[
\phi_{(u,ji)} = \frac{(ph_{ji,d} + 1)^F}{e^{m_{ji,d}}}, \tag{5}
\]

SARA uses greedy heuristics to restrain the number of paths that might be used to forward the traffic. The value of \( F \) is responsible for the traffic convergence on one route or a set of possible routes. The key aspect to consider here is the value used to define the convergence factor. If a low value of \( F \) is used, the cost tends to be the same, with slight differences in the pheromone level. The paths experience the same cost and thus a higher number of paths is maintained. Convergence on a smaller number of paths is then experienced when the value of \( F \) increases. By using an exponential function, the impact of shorter paths is stressed earlier. This avoids waiting for the time needed to increase the pheromone level of the corresponding path, which allows the shorter paths to be selected more often.

In SARA, all discovered routes are candidates for being used to forward the data packets, even those that have no active pheromones \((ph = 0)\). To do this, in Eq. (5), we increment the pheromone value by one unit and therefore every discovered path that remains active can be selected.

### 3.4. Route repair

The route repair is a process that is initiated when a broken link between two nodes is detected. The broken link state can happen due to a node being turned off, by failure in radio coverage or congestion that causes a higher number of collisions. In MANET, these kinds of situations may occur frequently and thus, the route repair procedure must be quickly executed with low overhead.

To repair the route, AntHocNet and SARA will try to find alternative routes in the neighborhood of the broken link. However, AntHocNet will attempt to reach the destination node with a standard FANT, but with a limited number of allowed broadcast transmissions, so that its proliferation is confined. SARA also uses broadcast transmission with R_FANT messages, but will attempt to find an alternative path that can reach the other side of the broken link instead of reaching the destination node. If several nodes go down simultaneously it might not be possible to find an alternative route, using this local repair procedure. If the local repair procedure fails to succeed, an error message is sent to the source node and the Route Discovery procedure is initiated.

To detect a broken link, SARA calculates a parameter, the \( \text{MAX}_\text{Tx} \), that indicates maximum transmission attempts which can fail before the link is considered broken or due to flow problems. The link quality is evaluated by two parameters, where \( \lambda \) is used every time a packet transmission fails and \( \delta \) when in the presence of a successful transmission \((\delta \) represents the recovery index of a link). A link presents transmission problems when the number of successful transmissions is lower than the number of unsuccessful ones. The link quality affects the model performance, so the packet transmission fail has a higher priority than the successful transmission. Due this, \( \lambda > \delta \in [0, 1] \). In these simulations, \( \lambda \) was set to 1 and \( \delta \) was tested with different values.

Let us consider a packet transmission that occurs at time \( t_i \), in link \((u, j)\). The link status can be given by Eq. (6), as follows:

\[
\text{NTx}_{(u,j,t_i)} = \begin{cases} 
\text{NTx}_{(u,j,t_i-1)} + \lambda & \text{if unsuccessful transmission,} \\
\text{NTx}_{(u,j,t_i-1)} - \delta & \text{if successful transmission.}
\end{cases} \tag{6}
\]

The minimum value admitted for \( \text{NTx}_{(u,j,t_i)} \) is 0. A link is considered with problems when:

\[
\text{NTx}_{(u,j,t_i)} > \text{MAX}_\text{Tx}.
\tag{7}
\]

SARA repairs a route through an incremental Deep Search Area (DSA) procedure proposed by us. The DSA is initiated by the node which detects the broken link. The DSA is a simplified procedure of Expanding Ring Search (ERS) \([31,32]\), with a TTL value of “2”. The ring search value of “2” is considered enough to find the other side of the broken link using a small amount of overhead. The value selected is within the values presented in the study of Jahan Hassan et al. \([31]\).

The DSA uses two types of messages: the Repair FANT (R_FANT) and the Repair BANT (R_BANT). The R_FANT is transmitted in a standard broadcast scheme, and the R_BANT in unicast back to the node that initiated the repair procedure. The node that initiates the Route Repair Procedure also initiates the Route Repair Timer (RRT). If the R_FANT reaches a node that knows how to forward the traffic to the destination node, it must reply with a R_BANT message through a unicast scheme. Should the route repair process fail, a RRT time out is issued and an error message is sent back to the source node stating that is not possible to re-establish the route. The complete route repair process is shown in Fig. 4.

### 4. Simulation studies

SARA is a routing algorithm which can be used in different environments with the right configuration of its working parameters. The simulation of different sets of values, assigned to these parameters, allows for the identification of how each one can affect SARA and how they can be tuned to work with different network conditions.

Thus, a first set of simulation studies (simulation set A) were used to evaluated the performance of SARA and to tune the parameters defined above, in order to reach an optimized behavior. The parameters were divided into major parameters and secondary parameters. The configuration of a major parameter can affect the behavior of SARA in accordance with the network state conditions and the
scenarios to be used. These parameters are responsible for a rough tuning of the algorithm. A secondary parameter allows for a fine tuning and its behavior is independent of the network conditions.

After completing the optimization of SARA’s parameters, the algorithm can be compared with others, in order to understand the impact of its original contributions to the network performance. Thus, in simulation set B, SARA was compared with the Ant Routing Algorithm (ARA), as it implements the basic ACO framework, the AODV as a classical solution and the AntHocNet. Both SARA and ARA were implemented in NS2 by the authors. The AntHocNet uses the code written by V. Laxmi, Lavina Jain and M. S. Gaur and presented in [33]. The comparison with a classical routing approach is also provided and for that purpose AODV was used with the code supplied with NS2 package version 2.31.

In both cases, the simulations were executed using the NS2 simulator, version 2.31. For each one of the tests proposed, 30 simulation runs were executed and a confidence interval of 95% was achieved. The 802.11b protocol was used for the MAC layer and the radios used the Two Ray Ground Propagation model and had a receiving range of 100 m and the data is 2 Mbps. The tests were done with random waypoint mobility model [34], where the nodes moved at the same speed. The node’s speed change between 0 ms⁻¹ and 10 ms⁻¹.

4.1. Simulation set A

4.1.1. Simulation scenario

Simulation setup consisted of 104 wireless nodes, placed randomly in a 1000 × 1000 flat space for 60 s of simulation time. In the first set of simulations, all the nodes were fixed in order to adjust the parameters.

Background traffic was established through the use of a low network load with one source node, until a medium network load level with four source nodes 4, of FTP/TCP type; the destination nodes were placed in the center of the scenario and the source nodes change from simulation run to simulation run. Each data packet had 1000 bytes of size.

The simulation procedure was aimed at tuning the following configuration parameters:

1. Convergence factor – \( F \)
2. FANT transmission rate – \( T_0 \)
3. FANT confirmation timer – \( T_1 \)
4. Route Repair Timer – \( RRT \)
5. Pheromone life time – \( \tau \)
6. Maximum transmission attempts – \( MAX_Tx \)
7. Link recuperation index – \( \delta \)

The evaluation of each one of these parameters was executed by using different values for the parameter under evaluation and keeping the other parameters fixed. This basic set of parameters defines the reference values, which have been empirically selected, and represents a realistic behavior of SARA, as depicted in Table 2.

For each one of the parameters, the comparison was made through the analysis of three distinct metrics:

**Number of used paths** – represents the total number of routes that are used to transmit traffic between each pair of source-destination nodes.

**Overhead** – it measures the amount of control traffic in relation to the total amount of traffic that has been sent.

**End-to-end delay** – represents the average difference between the receiving and the sending time of each packet.

![Fig. 4. Route repair procedure.](image-url)

#### Table 2

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Reference value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( F )</td>
<td>5</td>
</tr>
<tr>
<td>( T_0 )</td>
<td>100 ms</td>
</tr>
<tr>
<td>( T_1 )</td>
<td>100 ms</td>
</tr>
<tr>
<td>( RRT )</td>
<td>100 ms</td>
</tr>
<tr>
<td>( \tau )</td>
<td>1 s</td>
</tr>
<tr>
<td>( \delta )</td>
<td>1.0</td>
</tr>
<tr>
<td>MAX_Tx</td>
<td>5</td>
</tr>
</tbody>
</table>

Please cite this article in press as: F. Correia, T. Vazão, Simple ant routing algorithm strategies for a (Multipurpose) MANET model, Ad Hoc Netw. (2010), doi:10.1016/j.adhoc.2010.03.003
received packets. Lost packets are not taken into account.

During this procedure, it was observed that the convergence factor $F$ and the FANT transmission rate were the ones which could affect the model's performance more. These two parameters and how they work will be explained in more detail, being considered major parameters.

4.1.2. Study of the convergence factor – $F$

The convergence factor $F$ is used by SARA to converge the traffic into one route or to balance the load among multiple routes. This parameter is used in the calculation of the cost of a link, as shown in Eq. (5). The value of $F$ will turn the heuristics more or less greedy and the result is to force the traffic to converge into one route or to disperse in the network.

SARA was tested with different sets of values, from 0.5 to 10. Table 3 depicts the number of used routes of each convergence factor, under different network load conditions. The impact on the network performance is illustrated in the overhead and the end-to-end delay, which are represented in Fig. 5.

A global analysis of the results shows that SARA experiences higher overhead and end-to-end delay and uses less routes when the convergence factor is higher.

When the value of $F$ increases, the convergence capacity in the network also increases, as the traffic will tend to use the same path. When the network load increases (more sessions), because the traffic will flow through the same links, the probability of packet collisions will be higher. Due to this, more control packets will be required to repair the routes with excess of traffic or to discover new ones.

With a lower value of $F$, more routes will be available, which results in a network load balance and lower overhead. As more routes in the network are used, the link queues will present less traffic and the packets can flow in the network with more efficiency. So, the packets will travel faster in the network and the end-to-end delay will be lower, as shown in Fig. 5.

When the load increases, the use of multiple routes proves to be a better solution, so the value of $F$ should be lower.

4.1.3. Study of the FANT generation rate

FANT transmission is part of the Route Discovery procedure. The number of routes discovered in the network is related with the number of FANT agents generated and with the distance traveled by the FANT in the network. This distance is determined by the number of different hops crossed by each FANT during its activity period, which is calculated according to the following rules:

- the FANT arrives at the destination node,
- the node responsible to retransmit the FANT agent knows a valid route to the destination node, and
- the number of visited nodes by the FANT is lower than the maximum allowed (TTL).

The simulations were run with FANT transmission rate that varied from one FANT each 100 ms to one FANT each 5 s. The number of routes used are represented in Table 4 and the network performance in Fig. 6.

A global evaluation shows that the number of routes discovered is kept almost constant for all simulation runs. This unexpected behavior is due to the use of a convergence factor of '5' as a reference parameter, which implements a greedy strategy, leading to the usage of very few routes. The results obtained shows that the FANT transmission rate is not as critical as the value of $F$, specially when this one leads to the use of few paths.

Concerning the network performance, the results achieved have shown that, the overhead is affected by the FANT transmission rate, as more FANTs represents

Table 3
Convergence factor ($F$): number of used routes.

<table>
<thead>
<tr>
<th>N. sessions</th>
<th>$F$</th>
<th>0.5</th>
<th>1.0</th>
<th>2.0</th>
<th>5.0</th>
<th>10.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>65</td>
<td>32</td>
<td>8</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>81</td>
<td>45</td>
<td>21</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>134</td>
<td>62</td>
<td>33</td>
<td>18</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>143</td>
<td>80</td>
<td>39</td>
<td>24</td>
<td>24</td>
</tr>
</tbody>
</table>

Fig. 5. Convergence factor ($F$): network performance.
more control traffic. However, the end-to-end delay maintains a similar behavior during all the simulation runs: the use of a limited number of routes lead to an excess of traffic to the same routes and thus the end-to-end delay is almost always constant and relatively high.

### 4.1.4. Secondary parameters optimization

Apart from $F$ and the FANT transmission rate, that have a major impact on the behavior of SARA, the tuning of the remaining parameters enables an optimization of the model.

Different sets of simulations were realized, varying each parameter and keeping the others with the reference value. The value that offered the best performance was selected as the optimum value of the parameter under study. A final simulation with all parameters tuned to the optimum value was realized and the results that have been achieved are compared here with the results of the reference configuration (see Table 5). As the optimal value of $F$ is 1, this means that multiple routes will be used during all the simulation tests.

This set of tests comprises a broad range of perspectives and enables us to understand how to adjust SARA to different operating scenarios, including: number of routes used and its impact on the collisions; network performance (overhead and end-to-end delay) and data performance (data packets rate and data delivery rate).

#### Table 4

FANT generation rate: number of used routes.

<table>
<thead>
<tr>
<th>N.sessions</th>
<th>FANT TX rate ($T_0$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>19</td>
</tr>
<tr>
<td>4</td>
<td>23</td>
</tr>
</tbody>
</table>

![Graph](image)

**Fig. 6.** FANT transmission rate: network performance.

#### Table 5

SARA Optimum configuration values.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Optimum value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F$</td>
<td>1</td>
</tr>
<tr>
<td>$T_0$</td>
<td>500 ms</td>
</tr>
<tr>
<td>$T_1$</td>
<td>1 s</td>
</tr>
<tr>
<td>RRT</td>
<td>200 ms</td>
</tr>
<tr>
<td>$\tau$</td>
<td>500 ms</td>
</tr>
<tr>
<td>$\delta$</td>
<td>0.1</td>
</tr>
<tr>
<td>MAX_Tx</td>
<td>2</td>
</tr>
</tbody>
</table>

![Graph](image)

**Fig. 7.** Impact of the number of routes used.

Please cite this article in press as: F. Correia, T. Vazão, Simple ant routing algorithm strategies for a (Multipurpose) MANET model, Ad Hoc Netw. (2010), doi:10.1016/jadhoc.2010.03.003
The Fig. 7 shows SARA’s behavior when in the presence of multiple routes available in the network, by comparing the number of used routes with the number of collisions. When the traffic flows through different paths, the network load can be balanced and the number of collisions decreases. This can be an indicator of a better use of network resources and can improve the efficiency of the SARA model. However, for a low level of network load, the fast convergence into one best route in the network could be a good solution for the use of network resources.

The Fig. 8 shows the overhead generated and the path end-to-end delay. As the result of using multiple routes in the network, the number of route discovery and route repair procedures are lower and the overhead observed is also reduced. So, each node is able to use its resources to transfer the traffic in the network more efficiently and thus a slightly smaller delay is achieved.

The Fig. 9 shows the performance of data traffic when the optimum configuration is selected. When SARA uses a less greedy solution in the route selection equation, it is possible to experience alternative routes to reach the destination node. This way, SARA can balance the load in the network. When the convergence factor has a value of “1”, simulations run with the capability to use multiple routes. Due this, the data packets are spread across the network. The number of data packets delivered at the destination increases, but there is also the probability of the packets reach out of sequence at destination node.

In Fig. 9, it is also possible to observe the influence that the \( F \) parameter has on SARA to deliver data traffic to the destination node. As explained in Section 4.1.2, a higher \( F \) value can present better results with lower values of network load (higher path convergence) and a lower \( F \) is better when the traffic in the network increases (use of multiple routes).

4.2. Simulation set B

4.2.1. Simulation scenario

We also want to compare the performance of our protocol with other approaches, especially when the network is loaded and the nodes are moving. Thus, we perform a second set of simulations where we compare the performance of SARA (after the parameters were tuned to the optimal conditions) with the performance of ARA (based on the ACO framework), AODV (classical routing approach) and the AntHocNet. We consider a network with a medium load volume and different mobility scenarios, ranging from \( 0 \text{ ms}^{-1} \) to \( 10 \text{ ms}^{-1} \). Simulations at \( 0 \text{ ms}^{-1} \) were used as a baseline for comparison with the previous experiments.

In each simulation, all the nodes are configured with the same speed and the random waypoint mode [34]. The radio propagation range of the nodes is 200 m and the channel capacity is 2Mbps. Each simulation run for 60 s. The data traffic consists of four CBR sources sending eight 512-byte packets per second to four different destinations and four FTP clients accessing four different FTP servers. The FTP packet size is 1 kB. The tests with the CBR and the FTP traffic were done separately.

The simulation studies that were carried out aimed to evaluate and compare the performance achieved by SARA and by the other routing protocols under analysis.

The evaluation comprised four metrics:
Overhead – calculated by the ratio between the amount of information needed to carry control traffic over the total amount of traffic that has been sent.

Route discovery time – compares SARA CNB efficiency to discover a route with the other models.

Goodput – evaluates the quality of the routes and the capacity of the routing algorithm to maintain the data flow associated with an active session.

Packet delivery sequence – analyses the packet sequence variation at the destination node done by the use of multiple routes.

4.2.2. Study of the routing protocol overhead

The Fig. 10. presents the overhead results of the four routing protocols under analysis.

A global analysis shows that SARA presents the best performance, concerning overhead, which is almost independent of the node’s velocity. This fact is caused by the use of the CNB and by the optimizations that were realized in the route maintenance.

The use of a flooding mechanism and the continuous generation of control traffic to keep track of active routes are responsible for the worst performance of AntHocNet, AODV and ARA. The use of a source-routing mechanism, where all the entire path information is carried by the FANT messages is also responsible for the worst performance exhibited by AntHocNet.

These relative results are shared by UDP and FTP/TCP traffic. Nevertheless, smaller overhead values are experienced when FTP traffic is used, because the more significant amount of data traffic is generated.

4.2.3. Study of the route discovery time

The CNB model allows to find routes in the network with a reduced volume of overhead. However, with CNB working procedures, in which the network is flooded partially, the time required to discover a route could increase with the network load. To assess this, we simulate a medium loaded network on a high mobility scenario, where the nodes are moving at the highest speed considered (10 m/s). The results are shown in Fig. 11.

A global analysis of the results shows that all the protocols that use a flooding mechanism to setup a route need less time to setup a route and from this point of view, ARA, AODV and AntHocNet are more efficient than SARA, as most of the times they can discover a route to the destination node in <300 ms. With SARA, the CNB model will take more time to forward the FANT’s through the network, and so, the route discovery is higher. However, the route discovery procedure is called less times than the other models.

No significant difference arises when UDP or FTP/TCP traffic were used, as the four protocols present a similar behavior. Nevertheless, the number of routes discovered in a given period of time depends on the traffic that is being generated. Most of the times, less routes are discovered when FTP traffic is used, due to the congestion and flow control mechanisms of TCP.

4.2.4. Study of the goodput

The volume of data received at the destination nodes in a certain time period is related with the routing model’ ability to discover and maintain a route, as well as the path’ capability to transporting the packets through the network (link transmission time and node process time). Thus, the algorithm’ performance can be measured by the amount of information (data packets) delivered at the destination according to the period of time. The goodput also can be a metric which indicates the path quality during the simulation run time. The routing models behavior is presented in Fig. 12.

A global analysis of the results shows that SARA, ARA and AODV presents similar results when UDP traffic is used, which are better than the results achieved by AntHocNet. However, when the traffic is TCP, SARA has the best performance. ARA and AODV present similar results and AntHocNet also shows the worst results of all the protocols under evaluation. The reason for this behavior lies on the fact that, when the node’s mobility increase,
the network capability to maintain the routes decrease, and so, the transfer of data packets also decrease. In this scenario, SARA can route a larger volume of data packets than the other models. This larger value is associated with the amount of control traffic generated and the SARA’ capability to use the multiple routes. This situation is more visible when FTP traffic is used, because an higher throughput is used.

4.2.5. Study of the impact of using UDP or TCP traffic

To understand the different behavior experienced when UDP or TCP traffic sources were used, we also assess in detail these protocols. One key aspect in our analysis is the impact of losses or out-of-order packets, which certainly happen due to the collisions in the wireless medium and the possibility of using multiple paths offered by SARA.

In our study, we evaluate the sequence number of the received packets in order to detect sequence breaking. To assess the impact of multiple routes, we executed two simulation runs: one with SARA with a low convergence factor ($F = 1$, multiple routes) and another one with SARA with a high convergence factor ($F = 5$, very few routes). The results are shown in the Fig. 13.

For the CBR traffic, the network has the capability to transfer the data packets at the rate they are generated (eight packets per second per session). With this transmission rate, it is possible to observe the data packets reach the destination node without significant losses, as no significant missing sequence numbers were detected. In these conditions, SARA’ presents the same behavior as the other models.

With the FTP/TCP traffic the simulation results are different. SARA with a low convergence factor exhibits the best performance of all the protocols under analysis. The possibility of using multiple routes allows SARA to spread the data traffic and this avoids collisions, link congestion and packet drops.
5. Conclusions

This paper presents an improved version of the ACO framework, named as Simple Ant Routing Algorithm (SARA), that aims at reducing the overhead, by using a new route discovery technique, based on the concept of Control Neighbor Broadcast (CNB). The CNB allows SARA to control the control packets flooding level in the network. However, this flooding mechanism as the disadvantage of increase the time required to discover a route. In high mobility scenarios (node' velocity > 20 m s\(^{-1}\)) this could be a factor to decrease SARA’s performance and increase the overhead.

In the remaining phases, SARA further reduces the overhead by using implicit acknowledgments and by refreshing only the active routes. The proposed algorithm was described and compared with relevant work of the MANET’s routing research area.

The simulation results that have been presented show that SARA offers a good network performance and can be adjusted to different network conditions, based on an adequate selection of the major parameters (convergence factor and FANT transmission rate). Both parameters have impact on the number of used routes. According to the results achieved, using more routes enables load balancing and reduces overhead and collisions. Concerning the convergence factor \(F\), the results achieved have shown that small values of \(F\) are adequate for heavy loaded networks, while higher values of \(F\) present better results when the network load is lower. We intend to develop an algorithm that can dynamically adapt the convergence factor according to network traffic conditions. The tuning of the secondary parameters gives SARA into the best operating conditions, as stated by the simulation results achieved.

To finalize, the comparison of SARA with the classical ARA, the AODV and the AntHocNet demonstrates its main contribution: improving the network performance by reducing the overhead.

For future work, it is intended that an evaluation about the behavior of the pheromone variation will be performed and a correlation of these studies with the network state will take place. An experimental setup is also envisaged, using small and mobile robots.

References

Fernando Correia is currently a Naval Engi-
eneer of Electronic and Weapons System. He
graduated from the Naval Academy in Oc-
tober 1991. During his naval career, he
embarked on several ships, such as the NRP
"Honorio Barreto" from October 1991 until
October 1992. On board, he was the Electrical
Engineer Officer. During the period between
1992 and 1997, he was on board of the NRP
"Alvares Cabral" as Weapons Engineer Officer. During his assignments at sea, he
participated in several national and interna-
tional exercises, namely two STANAVFORLANT
NATO missions in 1993 and 1995. Ashore, in his area of expertise, he performed duties as Head of
Weapons Engineer Department at Gunnery Naval School.
During the period between September 1997 and September 1999, CDR
Fernando Correia did a Master of Science degree in Electronic and Com-
puter Engineering in Telecommunications, at Instituto Superior Tecnico in
Lisbon. In September 1999 he was appointed as Head of Fixed Commu-
nications of Telecommunications Division of the Portuguese Navy.
Since April 2004, CDR Fernando Correia is responsible for the mainte-
nance of Electronics and Weapons System, of the entire fleet of the Por-
tuguese Navy. During this period, he was an invited researcher for
Instituto de Engenharia de Sistemas e Computadores Investigação e
Desenvolvimento (INESC-ID) and he is currently doing a Ph.D in Com-
munication and Sensor Networks (WCSN), India, December 2006.
In parallel with his activities in the Portuguese Navy, CDR
Fernando Correia teaches Data Communication Networks, Linux Operat-
ing System and C/C++ Programming, at a Professional Institute.

Teresa Vazao is an Associate Professor at the
Department of Electrical and Computer Engi-
eering, at Instituto Superior Técnico, from
the Technical University of Lisbon, and is also
a researcher at INESC-ID. She has developed
research in the field of communications net-
works, focusing her work on the aspects of the
networks’ architectures. Currently, her
research interests are focused in routing prot-
ocols in Sensor Networks, Mobile Ad-Hoc
Networks and Vehicular Ad-Hoc Networks, as
well as Quality of Service Support. She pub-
lished more than 40 papers in conferences and journals. She has 2
national patents, 2 contributions to the IETF and she is a co-editor of 2
books. She has participated in several Portuguese and European projects,
as team member or coordinator of the Inesc-ID team’s activity. She is
Vice-President of Instituto Superior Técnico for the Taguspark Campus
Management since July 2009.