SAHR: Swarm Adaptive Hybrid Routing Protocol for Mobile Ad hoc Networks

B. M. G. Prasad\textsuperscript{1} and P.V.S. Srinivas\textsuperscript{2}

\textsuperscript{1}Associate Professor & Head, Department of Information Technology, Brindavan Institute of Technology and Science, Kurnool, Andhrapradesh, India – 518218. bmgprasad@gmail.com

\textsuperscript{2}Professor & Head, Department of Computer Science & Engineering Geethanjali College of Engineering & Technology, Cheeryal (V), Keesara (M), Hyderabad, Andhrapradesh, India

Abstract

Ad-hoc networks consist of autonomous self-organized nodes. Nodes use a wireless medium for communication, thus two nodes can communicate directly if and only if they are within each other’s transmission radius. Swarm intelligence refers to complex behaviors that arise from very simple individual behaviors and interactions, which is often observed in nature, especially among social insects such as ants. Although each individual (an ant) has little intelligence and simply follows basic rules using local information obtained from the environment, such as ant’s pheromone trail laying and following behavior, globally optimized behaviors, such as finding a shortest path, emerge when they work collectively as a group. In this paper, we apply this biologically inspired metaphor to the routing in mobile ad hoc networks. In this paper, we describe an algorithm which draws inspiration from Swarm Intelligence to obtain these characteristics. In an extensive set of simulation tests, we compare our routing algorithm with a state-of-the-art algorithm, and show that it gets better performance over a wide range of different scenarios and for a number of different evaluation measures. In particular, we show that it scales better with the number of nodes in the network.

Keywords: Manet, Swarm intelligence, hybrid routing, unicast routing, ACO
1. Introduction
Mobile ad hoc networks consist of a group of mobile nodes which autonomously establish connectivity via multi-hop wireless communications. Without relying on any existing, preconfigured network infrastructure or centralized control, they are useful in many situations where impromptu communication facilities are required such as battlefield communications and disaster relief missions. A number of ad hoc routing protocols have been proposed, for example, [1], [2], [3], [4], [5]. In proactive protocols such as [5], nodes in the network maintain routing information to all other nodes in the network by periodically exchanging routing information. Nodes using reactive protocols, such as [1], [2], delay the route acquisition until a demand for a route is made. Hybrid protocols, like [4], [6], use a combination of both proactive and reactive activities to gather routes to the destinations in a network – nodes using ZRP, for example, proactively collect routes in their zone, and other routes are collected reactively. In [6], on the other hand, the level of proactive activity and reactive activity are chosen autonomously by the nodes in the network, and proactive activity is only seen around favorite destination nodes. In most traditional reactive protocols, like [1], [2], only when a route breaks irreparably does the protocol mechanisms repair the damage. In reality, route deterioration is most often not sudden but gradual and most often available routes get better/deteriorate gradually and not suddenly. So the routing protocol should continuously maintain information about the nodes in the local area to perform effectively and avoid too many link breakages.

The rest of the paper organized as follows- Section II explores the most frequently cited related work, section 3 gives the overview of the proposed SAHR protocol. Section IV describes the root discovery and data transmission strategy in SAHR, section V explores the simulation and results discussion and that followed by the sections VI and VII contains conclusion and bibliography.

2. Related Work
Swarm systems have recently become a source of inspiration for the design of distributed and adaptive algorithms, and in particular of routing algorithms. Routing is the task of directing data flows from sources to destinations maximizing network performance. It is at the core of all network activities. Several successful routing algorithms have been proposed taking inspiration from ant colony behavior and the related framework of Ant Colony Optimization (ACO) [8]. Examples of ACO routing algorithms are AntNet [6] and ABC [19].

The ACO routing algorithms mentioned before were developed for wired networks. They work in a distributed and localized way, and are able to observe and adapt to changes in traffic patterns. However, changes in MANETs are much more drastic: in addition to variations in traffic, both topology and number of nodes can change continuously. Further difficulties are posed by the limited practical bandwidth of the shared wireless channel: although the data rate of wireless communication can be quite high, algorithms used for medium access control, such as IEEE 802.11 DCF[12] (the most commonly used in MANETs), create a lot of overhead both in terms of control packets and delay, lowering the effectively available bandwidth. The
challenges of autonomic control are therefore much bigger, and new designs are necessary to guarantee even the basic network functions.

3. SAHR Overview
1) When a route to a destination D is required, but not known at S, S broadcasts a Root Trace Swarm Agent RTSA to discover a route to D.
2) When D receives the RTSA from S, it initiates to transmit TRSA as Root Confirmation Swarm Agent RCSA, which transmits in backward manner through the path that traced by parent RTSA. The RCSA updates the routing table and secretion table of all the nodes in the path from S to D, allowing for data transfer from S to D. Here secretion table is maintained by each node \( n \) to store secretion attribute value \( \text{sav}_n \) of its each forwarding neighbor \( n_i \). The secretion attribute value is similar to pheromone repository of the biological swarm agent.
3) When a route fails at an intermediate node X then SAHR reinitiates root discovery process.
4) When a route at D is known to S, SAHR deterministically chooses the path by opting to best forwarding hop level neighbor \( n_i \) based on their hop level delay and number of hops to reach the destination.

4. Swarm Adaptive Hybrid Routing Protocol
SAHR’s style is galvanized by Swarm Agent Optimized routing algorithms for wired networks. It uses swarm agents that follow and update secretion tables in an indirect agent interaction for the modification of the surroundings learning method. Knowledge packets are routed stochastically consistent with the learned tables. A vital distinction with alternative Swarm Agent Optimized routing algorithms is that SAHR could be a hybrid algorithm, so as to deal higher with the precise challenges of Manet environments. It’s reactive within the sense that nodes solely gather routing info for destinations that they're currently communicating with, whereas it's proactive as a result of nodes try and maintain and improve routing info as long as communication goes on. we tend to build a distinction between the trail setup, that is that the reactive mechanism to get initial routing info a couple of destination at the beginning of a session, and path maintenance and improvement, that is that the traditional mode of operation throughout the course of a session to proactively adapt to network changes. The routing info obtained via indirect agent interaction is unfolded between the nodes of the Manet in hop level neighbor info exchange method to supply secondary steerage for the swarm agents. Within the following we offer a broaden description of the SAHR.

Description of the algorithm proposed
SAHR’s design is inspired by swarm agent optimized routing algorithms for wired networks. It uses swarm agents which follow and update secretion tables in an indirect agent interaction about the modification of the environment learning process. Data
packets are routed stochastically according to the learned tables. An important
difference with other Swarm Agent Optimized routing algorithms is that SAHR is a
hybrid algorithm, in order to deal better with the specific challenges of MANET
environments. It is reactive in the sense that nodes only gather routing information for
destinations which they are currently communicating with, while it is proactive
because nodes try to maintain and improve routing information as long as
communication is going on. We make a distinction between the path setup, which is
the reactive mechanism to obtain initial routing information about a destination at the
start of a session, and path maintenance and improvement, which is the normal mode
of operation during the course of a session to proactively adapt to network changes.
The routing information obtained via indirect agent interaction learning is spread
between the nodes of the MANET in a hop level neighbor information exchange
process to provide secondary guidance for the swarm agents. In the following we
provide a concise description of each of these components.

**Pheromone Indicator:**
Paths are implicitly outlined by the secretion tables that are kept regionally at every
node. An entry \( g_{ni} \) of the secretion table \( ST_i \) at node \( i \) that consider as pheromone
indicates about the goodness of the routing from node \( i \) to via immediate node \( ni \)
contains a price indicating the estimated goodness of going from \( i \) over neighbor \( ni \) to
reach destination \( d \). This goodness is derived from the combination of path end-to-
end delay and range of hops. These are commonly used quality measures in Manets.
Combining the number of hops with end-to-end delay between immediate node \( ni \) to
current node \( i \) and destination node \( d \) is a way to swish out presumably giant
oscillations within the time estimates gathered by the swarm agents. Since SAHR
solely maintains info regarding destinations that are active during a communication
session, and due to continuous change at neighbor nodes, the filling of the secretion
tables is dynamic.

**Route Discovery:**
The source node \( s \) determines the path to node \( d \) via broadcasting Root Trace
Swarm Agent \( RTSA \). At each neighbor hop that received \( RTSA \), broadcasts the same
to their neighbor hops. This process is recursive for each \( RTSA \) till it received by
destination node \( d \). Upon receiving the \( RTSA \), the destination node \( d \) initiates to
transmit Routing-path Confirmation Swarm Agent \( RCSA \) that derived from \( RTSA \).
\( RCSA \) Transmits in backward manner through the path that traced by parent \( RTSA \).
Upon reaching each node \( i \) in the routing path, \( RCSA \) updates pheromone indicator
value \( g_{ni} \) of relay hop node \( ni \) of the current node \( i \) in the routing path opted by \( RCSA \).
The process of updating the pheromone indicator value is as follows:

During the transmission of swarm-agent \( RCSA \), it collects the time \( t_{ni-ni} \) taken to
reach each node \( i \) from relay hop node \( ni \) the ‘ \( RCSA \)’ is coming from. The estimated
time \( t_{i-d} \) to transmit a data packet from node \( i \) to destination node \( d \) via
\( \{ni, ni+1, ni+2...ni+n\} \) is measured using equation (1).
And then pheromone indicator value will be measured using equation (2) and (3) that follow:

\[
t_{i \rightarrow d}^{ni} = (ni+n) \rightarrow d + \sum_{k=n}^{1} (ni+k-1) \rightarrow (ni+k) 
\]

(1)

Here in equation (3), \( h_{c_{i \rightarrow d}} \) indicates the hop count in path from current node \( i \) to destination node \( d \) via relay hop node \( ni \). The inverse value of the estimated time \( t_{i \rightarrow d}^{ni} \) for a data packet to travel from node \( i \) to destination node \( d \) indicates the optimality of the path between nodes \( i \) to destination node \( d \) via relay node \( ni \). Hence the equation (2) is significant.

Upon receiving swarm agent \( RCSA \), the source node \( s \) also updates its secretion table with pheromone indicator value \( g_{ni} \) of each neighbor hop \( n_i \) from the \( RCSA \) coming from.

**Data Transmission and Routing-path maintenance**

The routing-path maintenance will be carried out in proactive manner and will be initiated at destination node \( d \). The data transmission and path maintenance strategies explored in following subsections.

**Data Transmission**

In the process of transmitting data, source and hop level node selects the target neighbor relay hop dynamically. Initially source node finds best neighbor \( n_i \) based on pheromone indicator value of the nodes registered in its secretion table. Opting to a neighbor relay hop \( n_i \) with best pheromone indicator value \( g_{ni} \), transmits data packet to selected neighbor relay hop \( n_i \). Upon receiving the data packet the neighbor relay hop registers the sender’s information in routing cache. The strategy of selecting neighbor relay hop dynamically and transmitting data packet is recursive at each neighbor hop relay node. This process will be halted once the data packet received the destination node \( d \).

**Routing Path maintenance**

Upon receiving a packet \( dp_i \), the destination node \( d \) verifies the time \( t(dp_i) \) taken by \( dp_i \) to travel from source node \( s \) to destination node \( d \) and then measures the end to end delay for data packet \( dp_i \). If end to end delay of \( dp_i \) is exceeding the delay threshold \( \tau \) then it initiates a swarm agent \( RCSA \) and transmits towards source node.
that opts to the path accessed by data packet \( dp \). Hence the ‘RCSA’ performs the
process of updating pheromone indicator value \( g_{m} \) at each hop level relay node in the
path. This process explored in equations (1), (2) and (3).

Handling link failures
The destination node \( d \) initiates swarm agents RCSA to each neighbor relay hop
nodes in fixed time intervals. Hence the pheromone indicator values in secretion table
of each node will be updated in fixed time interval \( \zeta \).

The pheromone indicator value of any neighbor relay hop \( n_i \) in secretion table of
any node \( i \) is not valid if time since last update of \( g_{ni} \) is greater than time interval \( \zeta \).
This indicates the link failure between node \( i \) and destination node \( d \).

5. Experimental Results
The evaluation of SAHR was carried out in a number of simulation tests. The SAHR
performance was compared with the standard and state of the art protocol of the
Manet called AODV with local repair strategy[1]. As simulation software, we use
NS2. The protocol has been verified under various routing protocol evaluation
metrics. All of the test scenarios are obtained by varying parameters in a specific base
scenario. In this scenario, 100 nodes move in a flat, rectangular area of 3000 X 1000
m². Movement patterns are generated according to the random waypoint mobility
model (RWP) [13]: they choose a random destination point and a random speed,
move to the chosen point with the chosen speed, and rest there for a fixed amount of
pause time before they choose a new destination and speed. The speed is chosen
between 0 and 20 m/s, and the pause time is 30 seconds. Each simulation runs for 900
seconds. 20 different Constant Bit Rate sources start sending at a random time
between 0 and 180 seconds and keep sending till the end. At the Medium Access
Control layer, the IEEE 802.11b DCF protocol is used. As measures of performance,
we use the average end-to-end delay for data packets and the ratio of correctly
delivered versus sent packets. These are standard measures of effectiveness in
Manets. We also report delay jitter, the average difference in inter-arrival time
between packets. As measure of efficiency, we consider routing overhead, in terms of
number of control packets forwarded per successfully delivered data packet.

We investigate SAHR’s performance for varying levels of mobility and node
density, for increasing network sizes, and for different data traffic patterns.

To obtain scenarios with different levels of mobility, we vary the pause time.
Higher pause time means lower mobility, but also lower connectivity (due to specific
properties of RWP mobility, see [2]). The results are reported in figures 1 and 2.
SAHR shows much better effectiveness than AODV, in terms of average delay,
delivery ratio, and jitter. AODV has better efficiency, measured as routing overhead,
but the difference is rather small. The bad performance for high pause times is due to
the reduced connectivity.
Fig 1: Packet Delivery ratio comparison between AODV and SAHR

Fig 2: Packet overhead comparison between AODV and SAHR

Fig 3: End-to-End delay comparison between AODV and SAHR

Different node density levels are obtained by keeping the area size constant and increasing the number of nodes. The results of these tests are reported in figure 3. Again, SAHR performs better than AODV in terms of average end-to-end delay and delivery ratio, and the difference increases with the density. In terms of overhead,
SAHR is worse than AODV for low densities, but better for high densities. Jitter was not reported here, nor for the remaining tests, due to space limitations. It always follows more or less the trend visible for delay and delivery ratio.

For different network sizes, we increase the number of nodes (up to 800) and the area size together, keeping the node density constant. The results are presented in figure 4. SAHR’s advantage over AODV in terms of average delay and delivery ratio grows with the size of the network.

SAHR’s overhead grows less fast than that of AODV. This is an important result which indicates that SAHR is more scalable with respect to the number of nodes. For all the previous tests the data traffic consisted of 20 randomly placed CBR sessions. In figure 5 we show results of tests which use different traffic loads and patterns. We did tests with 20 and 50 sessions. The sessions are organized around a number of hot spots: 20 (or 50) randomly chosen CBR sources send to a limited number of different destinations. This number of destinations was increased from 1 up to the total number of sessions (corresponding to the randomly placed traffic we used before). Organizing traffic sessions around hot spots reflects the typical situation where traffic is concentrated around a number of important nodes. Again we observe an advantage for SAHR in terms of average delay and delivery ratio. This advantage is smaller for the easier scenarios where traffic is concentrated on a low number of hot spots. For the tests with 20 sessions, SAHR has higher overhead than AODV, while for those with 50 sessions the picture is more balanced.

6. Conclusion
In this paper we have described SAHR, a Swarm Adaptive hybrid routing protocol for MANETs which was inspired by ideas from Swarm Intelligence. The algorithm combines reactive and proactive behavior to deal with the routing challenges of MANETs in an efficient way. An efficient reactive strategy was explored for Routing path discovery. The data transmission and route maintenance was carried with an effective swarm agent based proactive strategy. The proactive strategy used in path maintenance is also equipped to help update pheromone on existing paths that helps to identify best path for further transmission and also helps to deal with link failures.

7. References

SAHR: Swarm Adaptive Hybrid Routing Protocol for Mobile Ad hoc Networks


