J. Indian Inst. Sci., Nov.–Dec. 2006, **86**, 553–568

An agent-based model for topology discovery and routing information fusion in mobile ad-hoc networks

S. S. MANVI* AND J. M. HURAKADLI
Electronics and Communication Engineering Department, Basaveshwar Engineering College, Bagalkot 587 102, India.
e-mail: agentsun2002@yahoo.com, sunil@protocol.ece.iisc.ernet.in, jyotimh2002@yahoo.com

Received on August 8, 2005.

Abstract

A mobile ad-hoc network (MANET) is a multihop wireless network where nodes communicate with each other without any pre-deployed infrastructure. The dynamics of wireless mobile ad-hoc networks as a consequence of mobility and disconnection of mobile nodes pose a number of problems in discovering topology and designing convergent and reliable routing schemes for effective communication between any source and destination. Therefore, topology discovery and routing information fusion issue is very important in the context of mobile ad-hoc wireless networks.

The proposed work uses an agent-based model to address the aspect of topology discovery and routing in mobile ad-hoc wireless network environment. Three kinds of agents are used: *Manager agent* (*MA*), *Monitoring agent* (*MOA*), *and Discovery and Routing agents* (*DRA*). MOAs of a node in a network are employed to monitor resources (transmit power, battery life, bandwidth, reliability, one-hop delay), whereas DRAs discover the links between the mobile nodes, perform routing information fusion and build pre-computed paths so that mobile users can communicate with each other based on the requirements (power/bandwidth/reliability/bandwidth and delay aware) of the network users. MA manages the activities of the discovery and routing agency. The model ensures network connectivity even under the conditions of uncontrolled mobility of nodes. Model has been simulated to test its operation effectiveness. It is observed that agent-based approach supports component-based software engineering features and facilitates flexible and adaptable discovery and routing information fusion services.

Keywords: Ad-hoc network, mobile agents, information fusion, power-aware, topology discovery.

1. Introduction

The living environment of the today's generation is emerging based upon information resources provided by the connections of various communication networks. New small devices like personal digital assistants (PDAs), mobile phones, handheld devices, and wearable computers enhance information processing and accessing capabilities along with mobility. A MANET (mobile ad-hoc network) is an autonomous system in which mobile nodes connected by wireless links are free to move randomly and often act as routers at the same time. Communication in MANETs is done either directly with one hop if they are within the range of each other, or indirectly using multihops through intermediate nodes by using the wireless link.

---

*Author for correspondence.

The most important features of MANETs are: autonomous terminals, distributed operations, dynamic network topology, multihop routing, fluctuating link capacity, resource-scare lightweight terminals. The typical applications of MANETs are in military battlefield, commercial sector, disaster areas and personal area networks. Regardless of the attractive applications, features of MANET introduce several research issues that must be studied carefully before a wide commercial deployment can be expected. The key issues concerning MANETs include routing, security, reliability, QoS (Quality of Service) and internetworking [1].

The efficiency of MANET depends not only on its control protocols, but also on its topology and routing information fusion. Topology is an important model of the network state as it implicitly gives lot of information about the active nodes present and the connectivity/reachability map of the system along with the resource information. Thus, it is desired to discover the topology and manage routing information. Topology discovery in MANET aims to discover the links between the mobile nodes and build paths so that any user can communicate with any other user. Topology management in ad-hoc networks is to ensure that the various network connectivity parameters are managed so as to ensure that the parameter values are within certain boundaries. It usually deals with: consuming less power/ energy, adaptable to node mobility, longer network lifetime, optimum path setup latency and increased network density. Based on the topology, routing information needs to be fused at every node, so that proper and reliable routes are computed.

Some of the related works are as follows. Power control is conceptualized as a network layer problem, and is presented so that protocol comprises of strategies to find an optimal transmit power to control the connectivity properties of the network, or a part of it [2]. Singh *et al.* [3] give some suggestions for the metric that include energy-consumed per-packet, time to network partition, variance in battery life of nodes, and the energy cost per-packet.

Some approaches aim at modifying the media access control (MAC) layer. Monk *et al.* [4] suggest to modify IEEE 802.11a handshaking procedure to allow nodes to transmit at a low-power level, while Singh and Raghavendra [5] propose enabling nodes to power themselves off when not actively transmitting or receiving. Xu *et al.* [6] focus on energy conservation by putting nodes to sleep using location information. Chen *et al.* [7], and Arabshahi and Gray [8] focus on local topology information obtained by using broadcast messages. A new class of routing algorithms based on principles of biological swarms is proposed by Arabshahi *et al.* [9], and Migas *et al.* [10], which have the potential to address problems like throughput, delay, energy and bandwidth constraints in an autonomous and intelligent fashion. An approach for topology management by making use of a set of PILOT (predefined, intelligent, lightweight topology management) nodes that can be used whenever a link between two sensor nodes is about to break is described in Srinidhi *et al.* [11].

A mobility-sensitive topology control method that extends many existing mobility-insensitive protocols is discussed and two mechanisms are introduced in Wu and Dai [12]: consistent local views that avoid inconsistent information, and delay and mobility management that tolerate outdated information. A distributed topology management algorithm that constructs and maintains a backbone topology based on a minimal dominating set (MDS) of the network is proposed in Bao and Aceves [13].

A detailed analysis of a cone-based distributed topology control algorithm is presented in Yelpern *et al.* [14]. Varying the transmission power at each node can control the topology of a wireless multihop network. Wattenhofer *et al.* [15] consider two degrees of freedom in topology management: the path setup latency and the network density. It uses sparse topology and energy management (STEM) technique that aggressively puts nodes to sleep. It provides a method to wake up nodes only when they need to forward data, where latency is traded off for energy savings. Multilayer mobility management architecture has been designed to take care of real-time and non-real-time traffic for intra-domain and inter-domain mobility in a survivable network [16]. An agent-based on-demand power and bandwidth aware routing for MANETs is proposed in Manvi *et al.* [17], which uses mobile agents to roam in the network by cloning at each of the selected node that supports power and bandwidth requirements until they discover path to destination. Manvi and Telsang [18] propose agent-based on-demand QoS routing for MANETs that discovers and fuses routes for an application based on bandwidth and delay metric.

For any kind of MANET application, it is desired to include the mobile nodes in the network connectivity map that have enough battery life so as to support the sessions. Power aware routes are required where ever it is needed to minimize the energy consumption to transmit to its neighbors. Bandwidth aware routes are required to support stored multimedia applications such as in the case of wireless distance learning. To support live multimedia applications such as in the case of disaster areas or battlefield communications, it is essential to have bandwidth and delay aware routes. Reliable routes are paths chosen such that the nodes on the path are more secured and do not go down frequently. By looking at all the kind of routes above discussed, we need to support flexible and adaptable kind of discovery services that can be selected by the group of users of MANET.

We observe from the literature that most of the works dealt with network topology, routing information fusion, and management does not provide flexible and adaptable services as described above. In our proposed work, we mainly use the agent technology to perform distributed topology discovery and fuse the routing information so as to discover the required routes and manage the network topology in MANETs, so that mobile users can communicate with each other based on the requirements (power/bandwidth/reliability/bandwidth and delay aware) of the network users that suit the applications. Agents are the autonomous programs situated within an environment; they will sense the environment and act upon it to achieve their goal. The agents can be static or mobile. A mobile agent differs from the static agent with respect to mobility. A mobile agent is a program that can migrate from node to node in a network of heterogeneous computer systems and fulfill a task specified by its owner.

The proposed scheme uses mobile agents that can move in the ad-hoc network to discover the topology and collect/update the routing metric such as transmit power, battery life, bandwidth information at every node as they move towards the targeted destination. At every node, mobile agent fuses the routing information and computes the routes. The static agent that runs in a host supplies the required information to the visiting mobile agent. Mobile agent precomputes routes based on requirements of the network users such as; least transmit power requirements and the battery life of the nodes, least transmit power and bandwidth availability, bandwidth availability and end-to-end delays, reliability and power requirements, etc.

The remainder of the paper is organized as follows. Section 2 gives brief introduction to software agents and describes the proposed model. Simulation model and its details are in Section 3. Section 4 presents simulation results. Section 5 gives the benefits of mobile agents and Section 6 concludes the work.

2. Proposed work

In this section, we explain in brief about agent technology and describe the proposed scheme for topology discovery and routing information fusion. The scheme assumes availability of an agent platform in the mobile nodes of ad-hoc network. However, in the case of nonavailability of agent platform agents perform traditional type of communication by using message exchanges to achieve their task. The agents use a knowledge base for inter-agent communication, which is based on blackboard architecture.

2.1. *Agent technology*

Agents are the autonomous programs situated within an environment, which sense the environment and acts upon it to achieve the goal(s). They have certain special properties, which make them different from the standard programs such as mandatory and orthogonal properties. Mandatory properties of the agents are *autonomy, reactive, proactive,* and *temporally continuous*. The orthogonal properties are: *communicative*, *mobile*, *learning* and *believable*. Mobile agent is an itinerant agent dispatched from source computer, which contains program, data, and execution state information, migrates from one node to another node in the heterogeneous network and executes at remote node until they accomplish their task [10].

An agent platform comprises agent server, agents, security manager, agent transport mechanism, and an agent execution environment. The services provided by an agent platform are: agent creation, security, persistence, agent execution, inter-agent communication and messaging. The mobile agent code should be platform independent, so that, it can execute at any remote node in the heterogeneous network environment. They communicate and cooperate with other agents to achieve their goals. Agent can update its information base while interacting with other agents during its travel. Inter-agent communication can be achieved by message passing, RPC (remote procedure call) or common knowledge base (black board). Some of the agent platforms are IBM Aglets, Odyssey, Grasshoper and Voyager [19, 20].

There are several good reasons for using mobile agents: reduce network load; overcome latency; encapsulate protocols; execute asynchronously (this feature is very much suitable for mobile ad-hoc networks); adapt dynamically and support component-based software development [21]. Network programmability is ultimate form of flexibility, which may open up design choices upward in layering closest to application, and later in time, the resulting functions may actually take place deep inside the network. The motivation for using the mobile agents is to perform dedicated processing inside network at particular links/clusters on behalf of customers/administrators, which may not implement required algorithm and services. A major incentive for mobile agent-based approach is that policies can be implemented dynamically based on resource states.

However, there are certain issues to be resolved in implementation of mobile agents such as agent transfer mechanisms, addressing, exporting agent state information, communication language, secrecy, privacy, agent data transfer, authority and portability. Security to mobile agents and nodes is an important issue as they visit different network servers. The host should be protected from agent and vice-versa from different types of attacks. Authentication credentials can be used to circumvent this problem [22].

2.2. *Agent-based topology discovery and routing information fusion*

Mobile agents have the ability to support asynchronous communication (need not require permanent connection between the parties) and flexible query processing. Therefore, the mobile user can assign a task to a mobile agent and whenever the agent feels that there is communication availability it will roam the network and fulfill the task delegated by its user. In this way, a mobile node requires less communication connectivity than the traditional client/server approaches. Another equally important reason for mobile agents in wireless networks is that they can reduce network traffic.

The mobile nodes running on battery power do not have enough power to discover topology and to run complex routing protocols necessary in ad-hoc networks. An alternative is to use mobile agents to discover topology by using routing metrics such as hops, bandwidth, power, delays, etc. and generate a routing table with that information based on the single metric or two metrics. And this table is used for proactive routing, thus saving battery life of mobile nodes due to asynchronous communication of agents.

The proposed model consists of topology discovery and routing agency. The agency at every node comprises a manager agent, a monitoring agent, discovery and routing agents, and a knowledge base for inter-agent communication. The model precomputes the paths based on the routing metrics required by the network users. Functional architecture of the model is presented in Fig. 1. Now we describe each component of the agency.

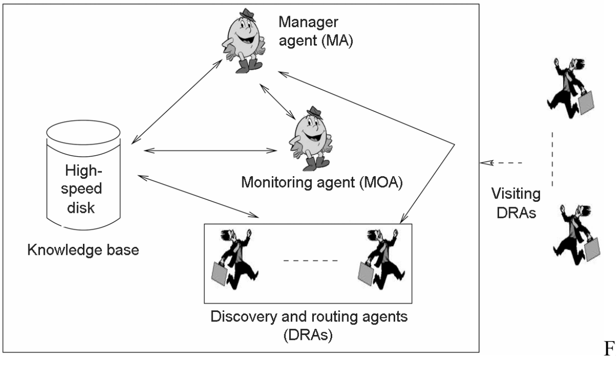


FIG. 1. Topology discovery and routing agency.

	1	2	3	4	5	6	7	8	
Neighbor-id	Battery maximum/ available (h)	Bandwidth maximum/available (Mbps)	Transmit power required (mW)	Reliability	Signal strength threshold (mw)	Battery threshold (min)	Delay (ms)	Prob. Loss	Node's info. & routing table pointer
12	3/2	10/2	2	1	0/5	10	0.57	0.01	Cols 1, 2, 4–8 and pointer to routing table

Ruting table based on transmit power

Destination	Next-hop	Routing-metric values (power)	Time of update
23	12	6 mw	11.30.05 am

FIG. 2. Agency knowledge base.

Knowledge base: The information about the neighbors, recent topology, and routing metric information is contained within it, which can be read and updated by the agents. It facilitates inter-agent communication. The parameters stored in the knowledge base of a node are: bandwidth information (maximum/available, battery values (maximum and remaining capacity) of its neighbors, transmit power required to neighbors, delays to neighbors, threshold signal strength to its neighbors (to detect the mobility of the neighbor, i.e. if a neighbor received signal strength falls below the threshold, it is treated as moving out of range), battery threshold value (if a node's battery capacity goes below threshold, it is considered as not suitable for long-lived multimedia applications), reliability status (node does not go down frequently) of a neighbor and a routing table (destination, next-hop, routing metrics, time of update). Figure 2 shows details of the contents of the knowledge base and routing table computed based on transmit power requirements. Routing table may also be formed based on combinations of other metrics as well.

Manager agent (MA): It is a static agent that is directly accessible to routing application protocol. It creates the following: (1) the knowledge base profile, (2) monitoring agents, and (3) discovery and routing agents. This agent synchronizes the activities of all other agents. Other agents either collect/update the knowledge base by seeking permission after getting authenticated from the manager agent. All the network users decide upon the type of routes required by using communication among the manager agents of the network nodes. Once the type of routes to be computed is fixed, MA creates discovery and routing mobile agent (DRA) code to do the required job. MA takes on three roles: (1) as a source, it initiates DRAs to find the routes based on the network user requirements, (2) as an intermediate node, receives DRAs and gets the path and other resources information from them as well as provides its routing and neighbors information, and allows DRAs to update the routing table, and (3) as a destination, receives the DRAs, exchanges information with them, allows DRAs to update the routing table, and disposes them.

Monitoring agent (MOA): It is a static agent that performs the following tasks at a node in which it resides.

- Monitors the neighbors by periodically exchanging hello packets.
- Updates the knowledge base with the monitored values by seeking permission from the MA.

- The parameters monitored are as follows.
  - Maximum and available network bandwidth in the neighboring nodes (we assume that certain amount of bandwidth is pre-allocated to the nodes in a cluster).
  - Battery capacities (maximum and remaining capacity) of its neighbors.
  - Transmit power required to neighbors based on the strength of received hello messages from its neighbors.
  - Delays to neighbors is computed based on the queuing delay at the node, the propagation delay and the number of retransmissions by observing the delays incurred in sending and receiving hello messages.
  - Fixes threshold signal strength to its neighbors (to detect the mobility of the neighbor, i.e. if a neighbor's received signal strength falls below the threshold, it is treated as moving out of range).
  - Battery threshold value of its neighbors (if a node's battery capacity goes below threshold, it is considered as not suitable for long-lived multimedia applications).
  - Reliability status (node does not go down frequently) of a neighbor, i.e. 0 = not reliable, 1 = reliable. Status = 0, if the neighbor does not go down for half the number of times, the node is observing the neighbor, otherwise 1.

Discovery and routing agent (DRA): This is a mobile agent, which migrates from one mobile node to another in a loop-free manner to discover the connectivity of the ad-hoc networks as well as fuse routing metric information at the visited node to update the routing tables. DRAs generated by MAS discover the nodes that are beyond one-hop. It collects the neighbors information of the visited nodes and maintains in its knowledge base and marks the nodes it has visited. Nodes marked will not be considered for migration, thus preventing loop movement of DRAs. The agent exchanges/updates the collected information base with the MA of the visiting node. It migrates to the unvisited neighbor node of a residing node that has least transmit power requirements and the battery life greater than the threshold value in case of power and battery aware route discovery. For power and bandwidth aware routing, it chooses a neighbor satisfying the following criteria: least transmit power, highest bandwidth availability with battery life above threshold value. In case of reliable and bandwidth aware routing, migrates to a node which has reliability status = 1, the highest bandwidth availability along with battery life above threshold value.

The next-hop at a node to reach a particular destination from a node will be updated by the visiting DRA. Path updating may be based on different kinds of routing metrics as agreed by the network users. DRA disposes itself, when it comes across a loop movement or if none of the neighbors satisfy the criteria of routing metrics. To avoid unnecessary wastage of bandwidth because of agent movement, agent migrations are limited to certain number of hops. In this way, when limited number of mobile agents are generated and made to migrate from one mobile node to another, the topology/reachability will be discovered. During migration, the DRA residing on a node provides the reachability information of the already visited nodes to the manager agent.

Algorithms 1 and 2 present the pseudocode for the topology discovery with power and battery aware routing and bandwidth and delay aware routing, respectively. In the same way we can also design schemes that can perform topology discovery and routing based on different metrics: transmit power and reliability, reliability and delays, battery power and reliability, bandwidth and loss probability, etc.

Algorithm 1: Topology discovery with transmit power and battery aware routing

Nomenclature: N = Number of mobile nodes, *nhop* = Number of hops an agent can travel, *num_dra* = Number of DRAs generated by each node, DRA_i = DRA generated at mobile node *i*. *visit_nodes* [..] = List of mobile nodes visited by DRA_i, where $1 \leq i \leq N$, *pow*[..] = is a list representing power required at a visited node from source (source is one which generates DRA).

```
Begin
1. For $i$ = 1 to N do
   Begin
   MA at node $i$ triggers MOA to discover the one-hop neighbors, their available battery power, transmit power requirements, bandwidth available and delays (propagation + queuing) by using HELLO messages either periodically or as and when need arises and store them in the knowledge base.
   End. //End of for $i$
2. For $j$ = 1 to N do
   Begin
   MA at node $j$ generates num_dra DRAs to roam in the network by initializing their knowledge base such as nhop = maximum hops, hop = 0, visit_nodes[..] = 0, pow[..]=0.
   End. //End of for $j$
3. For every $DRA_i$ generated by a mobile node $i$ at a regular time interval
   Begin
   $DRA_i$ being at node $i$ marks node $i$ as visited and collects $i$ node's information in terms of node-id, battery life, transmit power requirements from the node $i$'s knowledge base through MA, and sets visit_nodes[0] = $i$;
   • For hop = 1 to nhop do
     Begin
   – DRA at node $i$ will search for unvisited one-hop neighbor node with both a) battery life above threshold and b) least transmit power requirements among the set of one- hop neighbors of node $i$ using MA of node $i$ and its knowledge base (say, it is node $j$);
   – If $DRA_i$ fails in finding a node $j$, DRAi disposes itself at node $i$;
   – If $DRA_i$ is successful in locating a neighbor $j$ to migrate, collects the information of node $i$, migrates to $j$ and marks $j$ as visited; visit_nodes[hop] = $j$; hop = hop + 1; $i = j$, pow[$j$] = pow[$i$] + power required to migrate to node $j$;
   – At node $j$, DRA updates $j$'s routing table with the reachability information of all the nodes already visited by DRA by using MA of node $j$ and its routing knowledge base, i.e. it compares its pow values with existing pow values for the nodes it has visited; if less, then updates with new next-hop = $i$ to reach the visited nodes;
   End //End of for hop
End
```

```
4. If a node movement is detected by any of the mobile nodes by observing the received
   signal strength (received signal strength less than the threshold strength) or battery gone
   down, the node will send a broadcast message to all the nodes to repeat steps 2 and 3.
5. Stop
End.
```

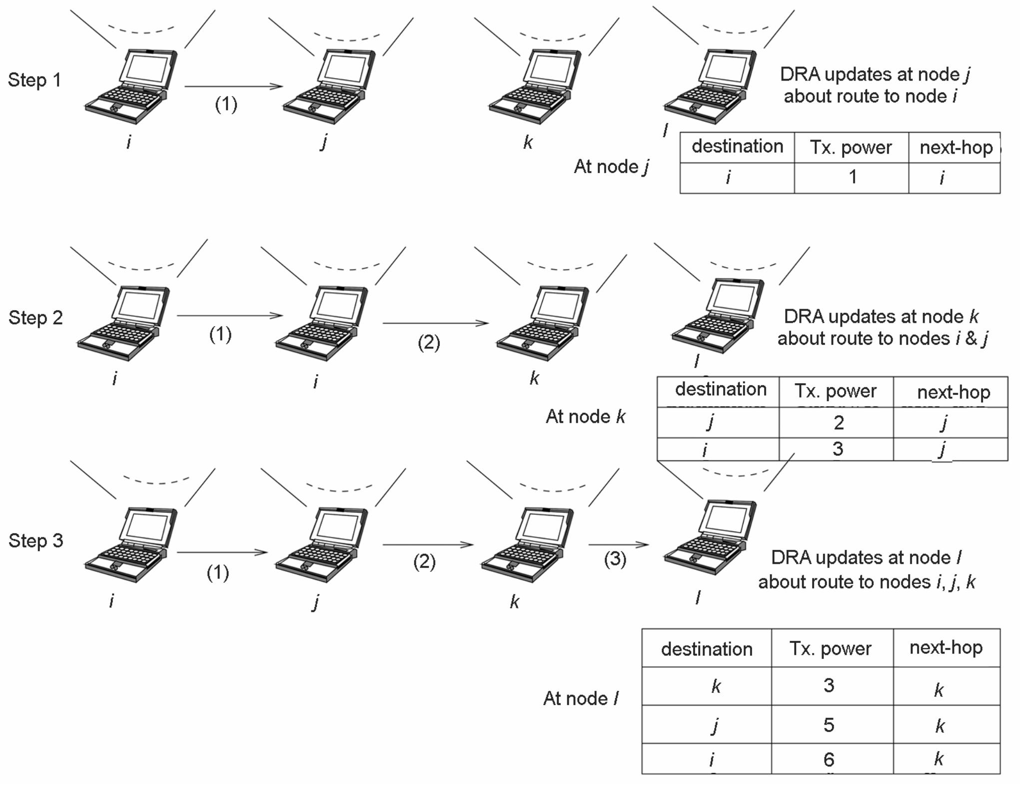
Algorithm 2: *Topology discovery with bandwidth and delay aware routing*

Nomenclature: Same as in algorithm 1 except *pow* is replaced by *delay*

```
Begin
1. Perform same as in step 1 of Algorithm 1
2. Perform same as in step 2 of Algorithm 1, but instead of using pow, it uses, delay metric,
   thus it initializes delay[..]=0;
3. For every $DRA_i$ generated by a mobile node i at a regular time interval
   Begin
   $DRA_i$ being at node i marks node i as visited and collects i node's information in terms
   of node-id, battery life, bandwidth, and delays from the node i's knowledge base
   through MA, and sets visit nodes[0] = i;
   •  For hop = 1 to nhop do
   Begin
   –  DRA at node i will search for unvisited one-hop neighbor node with both battery life
      above threshold and highest bandwidth availability among the set of one-hop
      neighbors of node i using MA of node i and its knowledge base (say, it is node j);
   –  If $DRA_i$ fails in finding a node j, DRAi disposes itself at node i;
   –  If $DRA_i$ is successful in locating a neighbor j to migrate, collects the information of
      node i, migrates to j and marks j as visited; visit_nodes[hop] = j; hop = hop +1; i = j,
      delay[j] = delay[i] + delay required to migrate to node j;
   –  At node j, DRA updates j's routing table with the reachability information of all the
      nodes already visited by DRA by using MA of node j and its routing knowledge
      base, i.e. it compares its delay values with existing delay values for the nodes it has
      visited; if less, then updates with new next-hop = i to reach the visited nodes;
      End //End of for hop
   End
   4. Perform the same step as in Algorithm 1.
   5. Stop
End.
```

2.2.1. *Example*

In this section, we illustrate with an example, topology discovery and routing information fusion by considering a topology as shown in Fig. 3 that comprises four mobile nodes represented by *i*, *j*, *k* and *l* with each node connected with several other nodes (not shown in figure). The values shown in the brackets of the arcs indicate the transmit power requirement. The proposed concept is described by using algorithm 1.



At node j

destination	Tx. power	next-hop
i	1	i

At node k

destination	Tx. power	next-hop
j	2	j
i	3	j

At node l

destination	Tx. power	next-hop
k	3	k
j	5	k
i	6	k

FIG. 3. Topology discovery and routing table updating by DRA.

The steps followed in discovering the topology by a DRA generated at node *i* is given below.

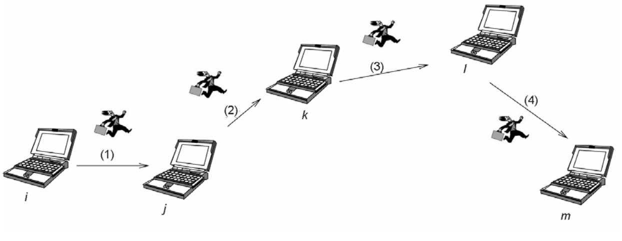
1. DRA is generated by node *i* and is initialized with nhop = 3 (number of hops agent can travel) and hop = 0 (a variable to keep track of hops agent is traveling).
2. At node *i*,
   - DRA collects information of *i* and marks it as visited. visit_nodes [0] = *i*.
   - DRA searches for the unvisited least transmit power node with battery power above threshold among the set of one-hop neighbors of node *i*. It is found to be node *j* with transmit power of 1 units.
   - DRA collects the information of node *j* from node *i*'s knowledge base, migrates to node *j* and marks it as visited; hop = 1 (= 0 + 1) and visit_nodes [1] = *j*.
3. At node *j*,
   - DRA updates the reachability information of the visited node *i*, i.e. DRA informs node *j*: to reach *i* from *j*, the next node is *i* and the transmit power is 1 unit.

- DRA searches for the unvisited least transmit power node with battery power above threshold among the set of one-hop neighbors of node *j*. It is found to be node *k* with transmit power of 2 units.
- DRA collects the information of node *k* from node *j*'s knowledge base, migrates to node *k* and marks it as visited. hop = 2 and visit nodes [2] = *k*.

4. At node *k*,
   - DRA updates the reachability information of visited nodes *i* and *j*. DRA informs node *k*: to reach *j* from *k*, the next node is *j* and the transmit power is 2 units, to reach *i* from *k*, the next node is *j* and the transmit power is 3 (= 2 + 1) units.
   - DRA searches for the unvisited least transmit power node with battery power above threshold among the set of one-hop neighbors of node *k*. It is node l with transmit power of 3 units.
   - DRA collects the information of node l from node *k*'s knowledge base, migrates to *l* and marks it as visited. hop = 3 and visit nodes [3] = *l*.
5. At node *l*,
   - DRA updates the reachability information of visited nodes *i*, *j*, and *k* at node l as follows: to reach k from *l*, the next node is *k* and the transmit power is 3 units; to reach *j* from 1, the next node is *k* and the transmit power is 5 (3 + 2) units; to reach *i* from 1, the next node is *k* and the transmit power is 6 (3 + 2 + 1) units.
6. Now, DRA has traveled 3 hops and hence disposes itself at node l.

Inter-agent communication takes through knowledge base at each of the nodes. Now we describe the inter-agent communication among DRAs by considering 5 mobile nodes represented as *i*, *j*, *k*, *l* and *m* as shown in Fig. 4 (values on the arcs indicate the transmit power requirements) to illustrate routing information fusion process.

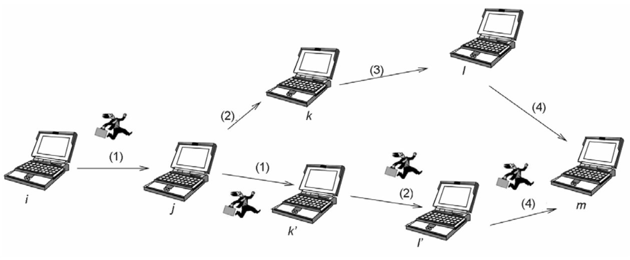
At time *t* = T1, DRA^{T1} is generated at node *i* and is initialized with nhop = 4 and hop = 0. DRA^{T1}1 migrates from node *i* to node *m* by updating the routing table knowledge base of nodes *j*, *k*, *l* and *m*. For instance, node *m*'s routing table knowledge base with the reachability information of the visited nodes is shown in Fig. 4.



Routing table updated by DRA at node *m* at *t* = T1

Destination	Tx. Power	Next-hop
l	4	*l*
k	7	*l*
j	9	*l*
i	10	*l*

FIG. 4. DRA movement from node *i* to *m* via *j*, *k* and 1 at time T1.



Routing table updated by DRA at node *m* at *t* = T2

Destination	Tx. Power	Next-hop
l	4	*k*
k	7	*l*
j	7	*l'*
i	8	*l'*
l'	4	*l'*
k'	6	*l'*

FIG. 5. Inter-agent communication between DRA^{T1} and DRA^{T2}.

After some time, say at time t = T2, two mobile nodes *k*' and *l*' move towards nodes *j* and *m*, respectively, and hence *k*' and *l*' (Fig. 5) are within the transmission range of nodes *j* and *m*. Thus, when an agent DRA^{T2} is generated at node *i* and initialized with nhop = 4 and hop = 0, it migrates from node *i* to node *m* via nodes *j*, *k*' and *l*', since path via node *k*' requires least transmit power from node *j* as compared to node *k*. Hence, reachability information is updated in the node *m*'s routing knowledge base (which was earlier updated by DRA^{T1}) (Fig. 5).

3. Simulation

The proposed scheme has been simulated on a Pentium-IV machine by using C++ programming language. In this section, we discuss the network model and simulation procedure used to test the proposed scheme.

3.1. *Network model*

An ad-hoc network of N nodes is generated by using random placement of the nodes within the area given by $w*z$ square meters. The transmission range of each node is denoted as R m and each cluster covers an area of 2R units from the cluster center. The transmit power required at a mobile node is denoted as *pow*. Threshold battery level is assumed to be 20% of the actual battery capacity; battery power available at a node in the network at any instant of time is uniformly distributed between 0.1 and 1.0 (1.0 means completely available). Threshold signal strength is denoted as *sst*. A node is assumed to generate *numa* number of DRAs to roam in the network to discover the topology. An agent can travel at most *nhop* hops.

The speed of movement of individual node ranges from x to y m/s. Each node starts from a random location and moves in any one of four directions—north, east, west or south—with predetermined uniform speed towards boundary. Once it reaches the boundary, it moves in the opposite direction. The number of applications that needs connection in the network is denoted as *napp*. The source and destination pair for the applications is randomly generated. The agent code size is denoted as s Kbytes. Agent processing and migration time is uniformly distributed between (a, b) ms.

3.2. *Simulation procedure*

To illustrate some of the results of simulation we have taken $N = 50$, $w = 1000$, $z = 700$, $x = 0$, $y = 20$, R is varied from 20 to 200 m, *nhop* from 2 to 14, *numa* from 2 to 14, *napp* = randomly generated from 50 to 100, *a* = 100 ms, *b* = 200 ms, *s* = 2 Kbytes, *pow* is randomly chosen between 0 and 99 units, sst = 15.

Simulation procedure is as follows:

Begin

1. Generate an ad-hoc network with given number of nodes.
2. Trigger the MA and MOA agents at the mobile nodes.
3. Trigger the DRAs at the nodes to discover topology periodically or whenever a need arises.
4. Compute the routes and prepare a next-hop routing table.
5. Generate applications with random source-destination pairs.
6. Compute the performance of the system.

End

The performance parameters measured are as follows.

- *Connectivity*: The connectivity of MANET is defined as the percentage of ratio of the number of reachable applications (that have the paths) to the total number of applications generated.
- *Power utilization*: It is defined as the percentage of ratio of total transmit power utilized for reachable applications to the total maximum transmit power available for reachable applications.
- *Agent discovery time*: It is defined as the time taken by DRAs to roam the ad-hoc network to discover the topology and fuse routing information (includes migration and execution times at each of the nodes).
- *Agent overheads*: It is defined as the average bandwidth required by a DRA during topology discovery and route information fusion, i.e. it is defined as ratio of bandwidth required by each DRA (size of agent multiplie by number of hops DRA travels) to number of DRAs generated.

4. Results

We observe from Fig. 6 that for 100 applications, percentage of connectivity rises with increase in number of DRAs generated and number of hops traveled by a DRA (denoted as HPA in Fig. 6). Connectivity of ad-hoc network ranges from 40 to 100% with different number of DRAs and hops/agent. 100% connectivity indicates complete topology discovery and availability of routing information at every node. The best choice of DRAs and HPA as observed from Fig. 6 is 8 DRAs and 12 HPA, respectively.

Figure 7 depicts the graph of power utilization for 100 applications. It is observed that the transmit power utilized increases with increase in number of DRAs and hops traveled by DRA. Figure 8 depicts the agent discovery time for different cases. We notice that time increases with increase in the number of DRAs and the number of hops traveled by DRA.

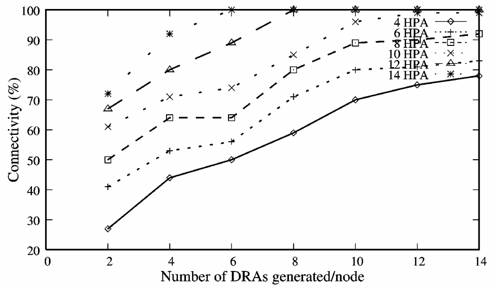


FIG. 6. Connectivity vs number of DRAs/node for 100 applications. HPA = Hops/agent.

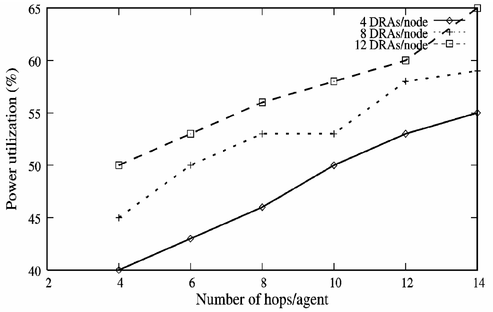


FIG. 7. Power utilization vs number of hops/DRA.

We observe from Fig. 9 that the agent overheads increase with increase in the number of DRAs and hops traveled by a DRA.

To test the bandwidth and delay aware routing we considered existing background load = 40% of cluster bandwidth, and generated 100 applications with bandwidth and delay requirements uniformly distributed between 200 and 300 Kbytes/s, 5 to 20 milliseconds, respectively, by considering cluster bandwidth = 20 Mbps and link+propagation delay varied between 400 and 800 microseconds. It is noticed that the results obtained showed less application rejections as compared to power and battery aware routing, since routes discovered were based on bandwidth availability.

5. Benefits of using agents

We experience that agent-based topology discovery and route information fusion offers flexibility, scalability, efficiency, adaptability, software reusability and maintainability.

Even though it is difficult to quantify these features, we explain below how they are achieved by using the proposed topology discovery and routing information fusion model.

Flexibility: The agents allow learning capabilities to be incorporated in a natural way for learning of node and network resources which helps in topology discovery and routing information fusion. The DRAs in our scheme can be programmed to even know the policies of the mobile nodes, reliability, prices, etc.

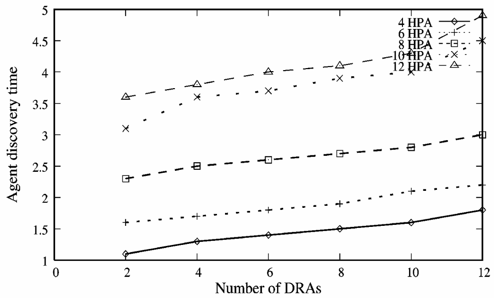


FIG. 8. Agent discovery time vs number of DRAs.

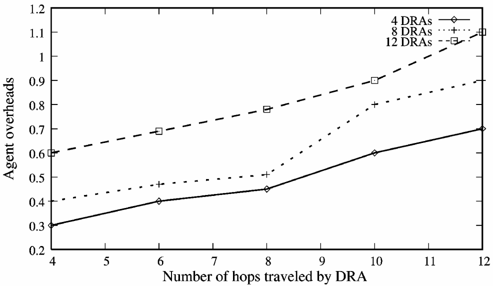


FIG. 9. Agent overheads vs number of hops traveled by DRA.

Scalability: The proposed scheme is scalable to any size of a network, since DRAs discover the topology in a distributed manner and are limited in traveling due to bound on the number of hops they can travel.

Efficiency: The scheme increases the efficiency of the network by discovering the topology based on battery power, transmit power, delays, link cost of the nodes. For example, the nodes which have less power do not get much load on them.

Adaptability: Functioning of the model itself justifies this feature. The nodes adapt to changed network conditions such as node movement and failures by sending the DRAs for discovering the recent topology and fuse route information.

Reusability: Reusability of the software can be seen in two categories: part of the mobile agent software and the algorithm software. The mobile agent can be reused for discovering resources within a network and create a local map of the resources. Algorithm software can be used for services discovery and services information fusion in case of m-commerce applications.

Maintainability: We can easily debug the agent components and also replace the old agent components with a new ones without affecting the other components.

Protocol encapsulation: The agents allow the network users to customize the protocol for their group activities and does not need any lengthy standardization process for protocol development as it is required in the current Internet protocol development.

6. Conclusions

Topology discovery and route information fusion based on different routing metrics and the network users requirements in ad-hoc networks is one of the challenging issues due to the highly dynamic nature of mobile nodes. In this proposal, we have presented a model for topology discovery and route information fusion in ad-hoc networks by using mobile agents. Mobile agents discover the links between the mobile nodes, perform routing information fusion and build pre-computed paths so that mobile users can communicate with each other based on the requirements (power/bandwidth/reliability/bandwidth and delay aware) of the network users. The model ensures network connectivity even under the conditions of uncontrolled mobility of nodes. It is observed that agent-based approach supports component-based software engineering features and facilitates flexible and adaptable discovery and routing information fusion services.

Proposed model can also be used for routing information fusion in Internet by considering different routing metrics such as bandwidth utility, delays, hops, cost and loss probability. Node movement within the MANET can be considered as node/link failures. By modifying discovery and routing agents, we can apply the proposed scheme to a heterogeneous network that comprises Internet and wireless networks (i.e. mobile Internet).

References

1. Jun-Zhao Sun, Mobile ad hoc networking: An essential technology for pervasive computing, http://www.mediateam.oulu.fi/publications/pdf/92.pdf

2. S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. R. Kumar, Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the COMPOW protocol, *Proc. European Wireless Conf.* (2002).
3. S. Singh, M. Woo, and C. S. Raghavendra, Power aware routing in mobile ad hoc networks, *Proc. ACM Mobicom*, pp. 181–190 (1998).
4. J. P. Monks, V. Bhargavan, and W. M. Hwu, A power controlled multiple access protocol for wireless packet networks, *Proc. IEEE INFOCOM*, pp. 219–228 (2001).
5. S. Singh, and C. S. Raghavendra, Power efficient MAC protocol for multihop radio networks, *Proc. Ninth IEEE Int. Symp. on Personal, Indoor and Mobile Radio Communications*, pp. 153–157 (1998).
6. Y. Xu, J. S. Heidemann, and D. Estrin, Geography-informed energy conservation for ad hoc routing, *Proc. ACM Mobicom*, pp. 70–84 (2001).
7. B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks, *Proc. ACM Mobicom*, pp. 85–96 (2001).
8. Payman Arabshahi and Andrew A. Gray, Intelligent routing and bandwidth allocation in wireless networks, *Proc. NASA Earth Science*, dsp.jpl.nasa.gov/members/payman/papers/estc 01.pdf (2001)
9. Payman Arabshahi, Andrew Gray, Ioannis Kassabalidis, Arindam Das, Sreeram Narayanan, M. El-Sharkawi and Robert J. Marks II, Adaptive routing in wireless communication networks using swarm intelligence, *Proc. 19th AIAA Int. Communications Satellite Systems Conf.*, Toulouse, France, April 17–20, 2001, http://dsp.jpl.nasa.gov/members/payman/papers/aiaa01-2.pdf
10. Nikos Migas,William. Buchanan, and Kevin A. McCartney, Mobile agents for routing, topology discovery, and automatic network reconfiguration in ad-hoc networks, *Proc. Int. Conf. and Workshop on Engineering of Computer Based Systems (ECBS)*, 2003, pp. 200–208. http://www.dcs.napier.ac.uk/ bill/wjbuchanan.pdf
11. T. Srinidhi, G. Sridhar and V. Sridhar, Topology management in ad hoc mobile wireless networks. http://www.cs.virginia.edu/ zaher/rtss-wip/38.pdf
12. Jie Wu, and Fei Dai, Mobility-sensitive topology control in mobile ad hoc networks, *Proc. 18th Int. Parallel and Distributed Processing Symp. (IPDPS)*, Santa Fe, New Mexico, April 2004. http://www.cse.fau.edu/jie/research/publications/Publication files/ipdps04-dai.pdf
13. Lichun Bao, and J. J. Garcia-Luna-Aceves, Topology management in ad hoc networks, *Proc. MobiHoc*, pp. 129–140 (2003). http://www.sigmobile.org/mobihoc/2003/papers/p129-bao.pdf
14. J. Y. Helpern, V. Behl, Y. M. Yang, and R. Wattenhofer, Analysis of cone-based distributed topology control algorithm for wireless multi-hop networks, *Proc. PODC*, Aug 2001, pp. 264–273.
15. R. Wattenhofer, G. D. Nguyen, and A. Ephermides. On the construction of energy-efficient broadcast and multicast trees in wireless networks, *Proc. IEEE INFOCOM*, March 2000, pp. 585–594.
16. J. Wu, and F. Dai, A generic distributed broadcast scheme for multi-layer mobility management in ad hoc wireless networks, *Proc. ICDCS*, May 2003, pp. 480–468.
17. S. S. Manvi, M. N. Kotabagi, and Bhanu Prasad, An agent based power and bandwidth aware routing for mobile ad-hoc networks, *Int. J. Uncertainty, Fuzziness Knowledge-based Systems*, **12**, suppl. 02, 83–100, (2004).
18. S. S. Manvi, and V. Telsang, An agent based approach QoS routing in mobile ad-hoc networks, *Proc. IEEE Int. Conf. on Signal Processing and Communications (SPCOM)*, Bangalore, India, 2004.
19. S. S. Manvi, and P. Venkataram, Applications of agent technology in communications: A review, *Computer Commun.*, **27**, 1493–1508 (2004).
20. D. Wong, N. Paciorek, and D. Moore, Java-based mobile agents, *Commun. ACM*, **42**, 92–102 (1999).
21. V. A. Pham, and A. Karmouch, Mobile software agents: An overview, *IEEE Commun. Mag.*, **36**, 25–37 (1998).
22. Danny B. Lange, and Mitsumi Oshima, Seven good reasons for mobile agents, *Commun. ACM*, **42**, 88–89 (1999).