

Review

A Review of Routing Protocols Based on Ant-Like Mobile Agents

Yasushi Kambayashi

Department of Computer and Information Engineering, Nippon Institute of Technology,
4-1 Gakuendai, Miyashiro-machi, Saitama 345-8501, Japan; E-Mail: yasushi@nit.ac.jp;
Tel.: +81-480-34-4111; Fax: +81-480-33-7461

Received: 17 June 2013; in revised form: 15 July 2013 / Accepted: 31 July 2013 /

Published: 6 August 2013

Abstract: A survey on the routing protocols based on ant-like mobile agents is given. These protocols are often employed in Mobile *Ad Hoc* Networks (MANET). Mobile *Ad Hoc* Networks are collections of wireless mobile nodes such as PDAs, laptop computers, and cellular phones having wireless communication capability that dynamically form a temporary network without using any existing network infrastructures such as wireless access points. The only infrastructure in MANET is the wireless communication interfaces on the devices. In such a circumstance, where some of the wireless devices are not within wireless range of each other, multi-hop routing is required to transmit messages to the destination. A node that wants to start communication with other nodes that are not within its one-hop wireless transmission range has to request intermediate nodes to forward their communication packets to the destination. In this paper, we survey a variety of proposed network protocols to accommodate this situation. We focus especially on biologically-inspired routing algorithms that are based on the ant colony optimization algorithm.

Keywords: MANET; ACO; routing P2P networks; mobile software agents; multi-agent system

1. Introduction

This paper presents a survey of routing protocols, based on ant-like mobile agents, which are intensively used in Mobile *Ad Hoc* Networks (MANET). MANET is a network that consists of wireless mobile nodes such as PDAs, laptop computers, and smart phones [1]. They have wireless communication capability that dynamically form a temporary network without using any existing

network infrastructure such as wireless access points connected to a wired network. The only infrastructure in MANET is comprised of the independent wireless communication interfaces on the devices. In such a circumstance, since the system lacks infrastructure such as routers, if some of the wireless devices are not within wireless range of each other, then multi-hop routing of data communication is necessary for messages to reach their final destination. A node that wants to start communication with other nodes that are not within its one-hop wireless transmission range has to request intermediate nodes to forward their communication packets to the destination. MANET is used in various applications including sensor networks and rescue and guidance systems in disaster situations [2–4].

Current routing protocols have certain intrinsic shortcomings. Since proactive routing policies such as Destination Sequenced Distance Vector Routing (DSDV) continuously updates the routing tables, they tend to waste network resources just for exchanging large amounts of routing table data [5]. It can be a serious problem because the network resources in MANET are assumed not to be robust. Too much internal activity may restrict the capability to transmit communication data for the actual users. On the other hand, reactive routing protocols such as Dynamic Source Routing Protocol (DSR) performs the route discovering procedures upon transmitting demands and may produce unacceptable delays in establishing the forwarding route [6]. In order to ameliorate such disadvantage in reactive routing protocol, *Ad Hoc* On Demand Distance Vector (AODV) was proposed as a result of integrating favorable features in DSDV into DSR [7].

Ant-like mobile software agents, which are analogous to the ones used in the Ant Colony Optimization (ACO) framework, are employed for discovering network topologies and thus efficient routing in the networks [8–12]. Ant-like mobile agents are an effective means to discover the network topology in particular in circumstances such as MANET in which the network topology frequently changes [9]. Unlike a proactive routing protocol, routing based on ant-like agents does not require frequent exchanges of update messages for routing tables. As the population of the network becomes dense, an ant-like agent becomes more effective for load balancing in the network. Ant-like agents are a known means to mitigate congestion [13].

2. Background of Mobile *Ad Hoc* Network Routing

Many routing protocols have been proposed for Mobile *Ad Hoc* Networks (MANETs) [1]. In this section we review two standard MANET routing protocols, *i.e.*, Destination Sequenced Distance Vector (DSDV) routing protocol and *Ad Hoc* On Demand Distance Vector (AODV) routing protocol.

2.1. DSDV Routing Protocol

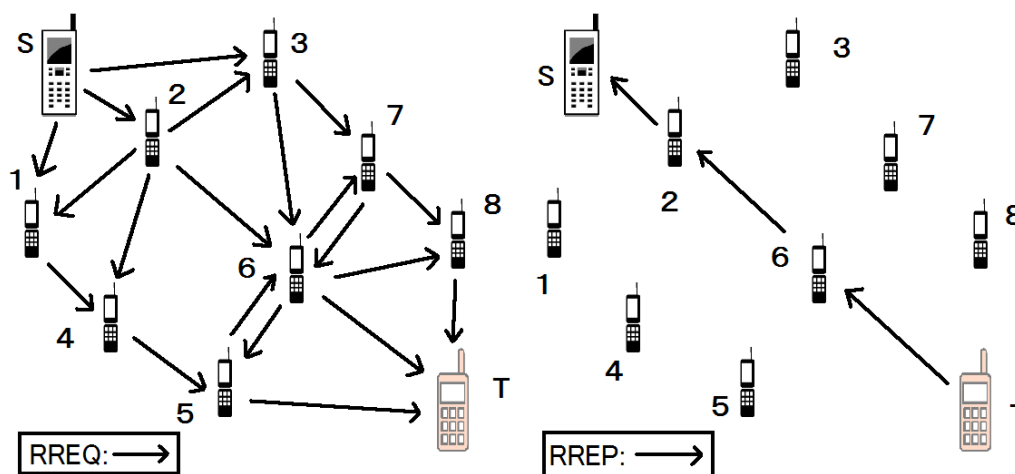
DSDV is based on the Bellman-Ford routing algorithm [5,14]. The MANETs that use this protocol try to make all the nodes store the routes to all the other nodes. Therefore, it requires the participating nodes periodically to broadcast maintenance information to keep the other nodes updated. Each node in the network has own routing table for all reachable nodes. Each entry of the routing table has corresponding sequence number and the number of hops to reach the destination node. The sequence number is used for identify obsolete routing information so that the algorithm can discard loops as well as diversions. During the updating process, each node periodically submits its own routing table to the neighbor nodes. Whenever the table is drastically modified, it sends the table to the neighbor nodes as

well. Each *route update packet* has a sequence number assigned by the source node as well as the routing table information so that the latest route information can be identified and used. When two different routes have the same sequence number, the shorter one is stored in the routing table.

2.2. AODV Routing Protocol

MANETs employing AODV routing protocol do not store routes to all other nodes. They maintain only routing information that is strictly necessary. Therefore, a participating node that wants to send a message to another node needs to find the route to the destination in the first place. When a node that uses this protocol wants to send a message to a destination node, and it does not know the route to the destination, it must perform the route finding procedure to identify the route to the destination [7]. The source node broadcasts the *route request packet (RREQ)* to its neighbor nodes. On receiving the message, the neighbor nodes further broadcast the message to its neighbors. These chains of broadcasting are continued until *RREQ* reaches the destination node, or *RREQ* reaches a node that has enough information to provide the route to the destination node. The left hand side of Figure 1 shows how *RREQs* move in the network. Unlike MANETs based on DSR use the source routing in which each data packet carries the complete path to the destination, MANETs based on AODV use on-demand routing in which each data packet has only next-hop information. Like DSDV protocol, AODV protocol avoids forming loops by using sequence numbers. Each node has its own sequence number as well as the broadcast identifier. The sequence number is incremented when the neighbor node is updated, and the broadcast identifier is incremented when the node performs the route finding procedure. In addition to these identifiers, the source node includes the sequence number to the destination node in *RREQ*. An intermediate node may have a larger sequence number than the one in *RREQ*. In such case, the intermediate node must know about the destination better than the source node, and therefore the intermediate node halts the *RREQ* and responds to the source node instead of re-broadcasting *RREQ*. Thus, in AODV, the data packet may have multiple paths to the destination, and can use the latest route to the destination.

Figure 1. *RREQ* and *RREP*.



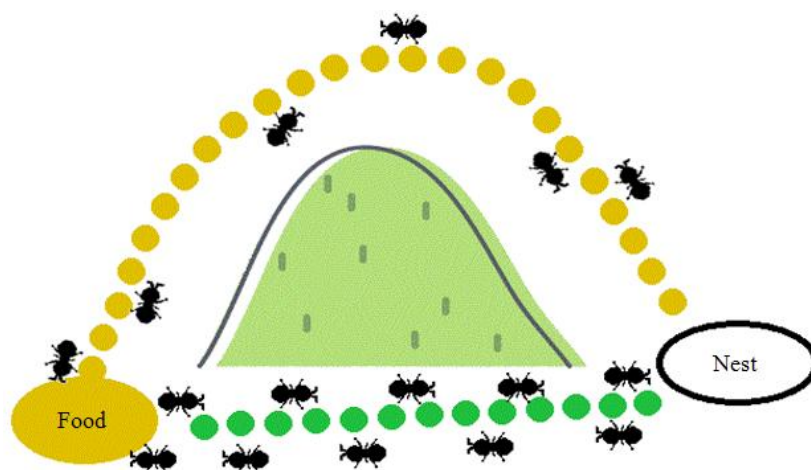
If *RREQ* reaches the destination node or an intermediate node that knows the route to the destination node, that node then responds to the source node by sending a route reply packet (*RREP*).

The right hand side of Figure 1 shows how *RREP* moves back in the network. In AODV, each mobile node periodically sends a *hello message* to other nodes to verify their existence. If a node finds that an intermediate node is lost and or that some routes are no longer passable, it broadcasts a *route error message (RERR)* to propagate the failure information.

3. Ant Colony Optimization

Ant-like mobile agents demonstrate their usefulness for solving some combinatorial optimization problems such as the travelling salesman problem [15]. The idea that network control software can use ant-like mobile agents to control communication network was introduced in late 1990s [13,16]. Today, various control methods based on ant-like mobile agents have been proposed for telecommunication networks [17,18]. The use of ant-like mobile agents for routing protocol was inspired by the fact that social insects like ants have a remarkable ability to solve complex problems in a distributed way. There is no central command. Individual insects act based on their local information, yet overall coordination is accomplished. Ants communicate with each other through an indirect method called stigmergy. Each ant lays down a special chemical substance in its environment that is read by other ants to identify it as having been at a given location. This feature has often been reproduced in ant-like mobile agent algorithms. However, social insect society also has the ability to recover from individual members' erroneous behaviors or deaths. This ability needs to be included into our artificial ant-like mechanisms. Deneubourg and Beckers demonstrated that ant-like agents can successfully solve complex problems and self-organize [19,20]. For example if there are several routes from the nest of ants to a food source, the ants may find some of them as shown in Figure 2. Every time the ants convey food from the source to the nest, they deposit pheromone so that the newly recruited ants can follow. When the ants deposit pheromone faster than it evaporates, the pheromone starts to accumulate. The more the ants follow a given path, the more the pheromone trail is strengthened. The ants may take the longer path as well as the shorter path. However, over time, the shortest route will accumulate greater pheromone density and it will attract more fellow ants. Such positive feedback works. Therefore, even though the ants find the shortest path later than longer ones, the shortest path eventually dominates. Deneubourg, Beckers, and other biologists, as well as computer scientists naturally derive the engineering application from this observation.

Figure 2. Ants take shorter path after initial search phase.



For example, consider solving the traveling salesman problem (TSP) using the artificial ant colony approach. Each ant builds a solution by using two types of information locally accessible: (1) problem-specific information, *i.e.*, distance among cities, and (2) information added by ants, as pheromone values, during previous iterations of the algorithm. While building a solution, each ant collects information on the problem characteristics and on its own performance, and uses this information to modify the representation of the problem. The modified representation is the new environment shared by all the ants and seen locally by other ants. The representation of the problem is modified in such a way that information contained in past good solutions can be exploited to build new better solutions. This form of indirect communication mediated by the environment is called *stigmergy*, and is typical of social insects [21–24].

Suppose an ant k (in m ants) is at city i , and N^k is the set of k 's unvisited cities. Then, the ant chooses city j to visit next in a probabilistic way as follows:

$$p^k(i, j) = \frac{[\tau(i, j)]^\alpha [\eta(i, j)]^\beta}{\sum_{l \in N^k} [\tau(i, l)]^\alpha [\eta(i, l)]^\beta} \quad (1)$$

here, $\eta(i, j)$ represents the problem-specific information, and in TSP, it is the reciprocal of the distance between two cities i and j . α and β are non-negative constants, and their values are determined by whether one attaches importance to global information or local information. $\tau(i, j)$ represents the information added by ants, *i.e.*, the pheromone value on the route between two cities i and j . This value is updated in the following rules, when one iteration is done.

$$\tau(i, j) \leftarrow (1 - \rho)\tau(i, j) + \sum_{k=1}^m \Delta\tau^k(i, j) \quad (2)$$

$$\Delta\tau^k(i, j) = \begin{cases} 1/L^k & \text{if } (i, j) \in T^k \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

here, T^k represents the set of all routes from the start city to the goal city, and L^k represents the collective distance from the start city to the goal city. In ant-colony algorithms, the pheromone value added to a specific route is determined by the reciprocal of the total distance and the pheromone also disappears over time, where ρ represents the evaporation rate so that newly added pheromone has more influence than old pheromone additions. Thus, after many of iterations, one can expect to obtain quasi-optimal route selection for a traveling salesman.

4. Routing Protocols Using Ant-Like Mobile Agents

There are many network routing applications using ant-like mobile agents. The ant-like agents are rather simple ones, moving from one node to another to collect connectivity information, and update the routing table at the each node.

In network routing applications such as [9,16], ant agents accumulate the information about the connectivity of the nodes in the network, and each time they visit a new node, they update the routing table based on the accumulated information. The volume of the information is not necessarily huge. The more they accumulate, the more the overhead they have to handle. A balanced design between information and overhead is important. All the nodes in the network depend on the information

provided by the ant-like agents, and they do not seek its own route finding activity. In this section, we discuss several routing algorithms that use ant-like agents.

4.1. AntNet

AntNet uses two types of ant-like agents for routing in the communication networks. They are the *forward ant* and the *backward ant* [25–27]. AntNet uses the *forward ants* to explore the network topology proactively. Each network node launches ant-like agents, *i.e.*, the *forward ants*, asynchronously toward randomly selected destination nodes. This happens concurrently with the data traffic at regular intervals. Each agent searches for a minimum cost path joining the source node and the destination node. While moving, the ant-like agents collect information about the time length and the congestion status of the followed paths. When the agents arrive at the destination node, they create the other ant-like agent, *i.e.*, the *backward ants*, and make them go back to their source nodes through the same paths as they come but in the opposite direction to reflect the traffic statistics that the *forward ants* collected. During the backward travel, the ant-like agents update the routing tables in the nodes along the paths with information they have collected during the travel. Thus, they adaptively build routing tables [25–27]. Di Caro *et al.* successfully showed the ability of artificial ants to optimize the network configuration. By using two different types of ant-like agents, *i.e.*, the *forward ant* and the *backward ant*, each node in the network has shortest path to all the destinations. However, they cause extra overhead and reduce the available traffic capacity for actual data communication. AntNet is an algorithm for traffic adaptive routing for multiple paths in wired IP networks. The effectiveness of this algorithm is demonstrated in the work of Schoonderwoerd *et al.* [28,29]. They have applied the algorithm for relatively static telephone networks, and the effectiveness in a network that dynamically changes its topology, such as MANET, may be limited.

Since the main purpose of AntNet is to investigate how data packets travel along a certain route, the *forward ants* and data packets share the same queue in the intermediate nodes. Thus the *forward ants* can measure the estimated travelling time that the actual data packets would experience. The problem, however, is that the *forward ants* may take a significantly lengthy travelling time to reach their destinations. In order to mitigate this problem, Di Caro *et al.* developed an improved AntNet, AntNet-FA. In AntNet-FA, *forward ants* also use high-priority queues in order to try to reduce the travelling time to their destinations [30]. As *backward ants* simply update the routing tables at the intermediate nodes based on the *forward ants*' experience, they need not measure anything, and they use high-priority queues in both AntNet and AntNet-FA. The *forward ants* in AntNet-FA estimate the virtual trip time based on the properties of the link such as bandwidth and propagation delay, instead of measuring actual trip time. AntNet-FA has improved AntNet with respect to efficiency, but it does not help it adapt to dynamically changing networks such as MANET.

4.2. AntHocNet

In a MANET, nodes are mobile and therefore enter and leave the network frequently. It is not reasonable or useful to maintain a complete description of the network. In order to cope with such situations in MANET, the same research group of AntNet has developed an algorithm that builds and maintains routing tables reactively with some proactive actions to refresh the routing information. In a

sense, AntHocNet can be seen as the MANET version of AntNet-FA. AntHocNet is a routing algorithm that pursues multiple paths simultaneously [31–34]. As in AntNet and AntNet-FA, ant-like agents roam in the network seeking routing information; however they are dispatched reactively when a source node wants to start communication with the destination node. The quality of a path is quantified as the inverse of the cost that is the number of hops travelling from the source to the destination, and it is expressed as the pheromone value. When a source node wishes to start a communication session to a destination node, the source node checks whether there is a route from the source to the destination. When there is one, the source node simply uses it; when there is not, the source node must discover a route to the destination. The source node creates an ant-like agent called *reactive forward ant*, to explore the network to seek the route. The *reactive forward ant* collects the route information as well as looking for the route. When an ant agent reaches the destination node, it becomes the *backward ant*, and travels back to the source node. Along the path, the *backward ant* updates the routing tables based on the route information collected by the *reactive forward ant*. Data packets are probabilistically distributed into several paths. This is accomplished by probabilistically selecting next node to transmit based on the pheromone value at each intermediate node.

Even the transmission route is established from the source node to the destination node and the data communication session started, the source node does not remain passive. It creates other ant-like agents called *proactive forward ants* and distributes them into the network. These *proactive forward ants* move from node to node based on the pheromone value, and monitor the quality of the route. From time to time, an intermediate node broadcasts additional *proactive forward ants* to explore new routes to the destination.

There are a few improvement proposals to AntHocNet. A notable one is Fuzzy Logic Ant Based Routing (FLAR) proposed by Mirabedini *et al.* [35]. FLAR has integrated fuzzy logic into AntHocNet. In MANET environments, route failure management is a challenge because of the node mobility and rather narrow bandwidth. Mirabedini *et al.* have proposed an adaptive fuzzy ant-based routing algorithm that combines ACO and the fuzzy logic technique to cope with the route failure problem. A similar proposal can be found in [36]. They have compared average end-to-end delay and packet delivery ration with AntHocNet, and have shown the effectiveness of the integration of fuzzy logic to ACO.

4.3. Ant-Colony Based Routing Algorithm (ARA)

One of the representative reactive routing algorithms based on ant-like mobile agents is Ant-Colony Based Routing Algorithm (ARA) [37]. ARA is a purely reactive MANET routing algorithm. One of its notable characteristics is that it does not use the *hello packet* to explicitly find its neighbors [38]. Instead, ARA uses a flooding technique. The source node in a MANET that employs this algorithm broadcast *forward ants* toward the destination. As each *forward ant* moves toward the destination, it constructs a route from the visited node to the source node. In ARA, a route is indicated by a positive pheromone value in the intermediate nodes' routing tables. When the *forward ant* reaches the destination, it creates the *backward ant* and sends it back to the source node in order to construct there the correct forward route. Since the all *forward ants* that share the same destination from the same source node also share the same sequence number, redundant ants returning via inferior routes are

simply discarded. Superiority and inferiority are determined by the pheromone values that are computed based on the number of hops of the route.

As in AntHocNet, the quality of a path is managed by a pheromone value, but it is maintained by data packets instead of ant-like agents. Each time an intermediate node relays a data packet along a certain path, the corresponding pheromone value is increased. As the pheromone evaporates, the less utilized route has less pheromone value and does not attract data packets.

When an intermediate node leaves from a route while the route is still used in a communication session, the node that failed to relay data sends an error message to the previous node. On receiving the route failure message, the previous node checks whether there is an alternate route to the destination. If there is one, the previous node simply uses it; if there is not, the previous node asks neighbors to relay the data packet. If the failure notification reaches to the source node, it re-initiates the broadcast of the forward ants.

Gunes *et al.* have demonstrated the effectiveness of ARA over AODV [39]. They have shown that the higher the mobility of the participating nodes in the network, the greater the effectiveness of ARA. Arif *et al.* have improved ARA in order to cope with much higher mobility situations. They propose an improved algorithm called Enhanced Ant Colony Based Routing Algorithm (EARA) [40]. In ARA, the pheromone values are computed by the number of hops, and delay time is not taken into consideration. In EARA, the initial pheromone value is set by not only the total number of hops made by the *forward ant* that pursued the route, but also by taking account of the time interval between the sending of the *forward ant* and receiving the *backward ant*. The inclusion of time period in calculating pheromone values creates a pheromone gradient from the source node to the destination node where only hop count produces similar gradients for entire route. This difference creates the higher delivery ratio [40]. Cauvery *et al.* have also improved ARA by handling loss of ants and avoiding re-transmission of lost packets by reserving resources at intermediate nodes [41].

Woo *et al.* have proposed a routing protocol for MANET similar to ARA [42]. In the protocol, the source node establishes the route by using reactive ant-like agents. After establishment, it creates *proactive ants* and sends them to explore the network in order to discover better routes. When an intermediate node has multiple routes to the destination, it selects the transmission path probabilistically and send data packet along it. This yields better balance of data traffic. If an intermediate node finds routing failure, the intermediate node creates reactive ant agents and dispatches them toward the destination.

4.4. Mobile Agent Routing (MAR)

Zhou and Zincir-Heywood proposed an algorithm called Mobile Agent Routing (MAR) [43]. In MAR, each data packet is considered as an agent that explores the network to collect routing information. Like ARA, MAR protocol has a certain number of ant-like agents, and they constantly explore the network to collect the routing information [43]. The ant agent of MAR has a history list with fixed length. The list contains the intermediate nodes the ant agent has visited. When the ant agent arrives at a node, the node looks at the history list, and if it finds better routing information than one in its own routing table, the node copies the information from the list to the routing table. If the node has better information than one in the history list of the ant agent, it also copies it to the history list of the

ant agent. Then the node lets the ant agent move to a randomly selected neighbor node. If the nodes in the network come and leave frequently, the number of ant agents is increased. When the network becomes stable, the number of ant agent is decreased. In order to assess the stability of the network, each node sends *hello messages* to its neighbors. If the node finds no reply from a certain node, it regards the node as having left the network, and deletes its entry from the routing table.

4.5. Probabilistic Emergent Routing Algorithm (PERA)

Probabilistic Emergent Routing Algorithm (PERA) uses three types of ant-like mobile agents [44]. They are the *forward ant*, the *uniform ant*, and the *backward ant*. Each node periodically creates *forward ants*, and sends them to randomly selected destinations in the network. Each intermediate node transmits the forward ants to the next node towards the destination based on the probability of reaching the destination. Since the next node is selected from the entries in the routing table in the intermediate node, this method fails to find new routes. In order to promote the adaptability of the algorithm, the participating nodes create the *uniform ants* and send them to explore the network. The *uniform ants* are transmitted to the neighbor nodes with uniform probabilities. The number of *uniform ants* is much smaller than the *forward ants*.

When the *forward ant* or the *uniform ant* reaches the destination, the *backward ant* is created and transmitted to the source node. The *forward ant* or the *uniform ant* hands the history list to the *backward ant* so that the *backward ant* can trace the route the *forward ant* or the *uniform ant* came along. While following the route, the *backward ant* updates the routing tables of the intermediate nodes based on the information in the history list.

4.6. GPS/Ant-Like Routing in Ad Hoc Networks (GPSAL)

This routing algorithm, GPS/Ant-Like Routing in *Ad Hoc* Networks (GPSAL), assumes all the mobile nodes participating in the network possess GPS (Global Positioning System) capability [45]. In the routing table of each node in the network, the participating nodes are listed with the current positions and time stamps. The table also shows whether a certain node is mobile or not. When a new comer wishes to take part in the network, it listens to the wireless data to find a member. On discovery, it sends its own routing table to the node and asks to be registered. The route from the source node to the destination is found by a shortest path algorithm based on the physical locations. The routing information in the intermediate node is copied to the data packet with time stamps. If a node does not have the route to its destination, it asks the closest non-mobile node to forward its request for a route to the destination.

Ant-like agents are used to update the routing tables of the network. They do this proactively. Similar to the other ant-based algorithms, they are sent to randomly selected destination nodes. When a node receives the ant agent, it compare the routing information in the ant agent with the one in the routing table, and copies the better information to the routing table as well as the history list of the ant agent. To explore the network, the ant agents select their next hop from the oldest entries in the routing table or the physically farthest nodes in the network. Once the ant agent arrives at the destination, it is made return to its origin.

4.7. Ant-AODV

In order to ameliorate the dilemma between proactive routing protocols and reactive routing protocols, it is natural to conceive of AODV routing protocol that does not consume network resources as DSDV does but integrate proactive ant-like mobile agents. Marwaha *et al.* have proposed such routing protocol called Ant-AODV hybrid routing protocol [46]. As the base protocol is an on-demand type, when the source node does not have the route information to the destination, it starts an on-demand route discovery procedure. It provides, however, proactive mobile agents, and they update the routing tables of participating nodes so that they decrease the frequency of the on-demand route discoveries. In such a compromise, it is most important to discover the appropriate parameters based on intensive simulations reflecting the real situation. In real world, constructing a hybrid system based on real compromise data may produce good enough MANET.

4.8. Multi-Agents Based Models

Using ant-like agents for route discovery is not very effective for MANETs. In order to cope with the frequent changes of the network topology, a large amount of control data must be exchanged as ant-like agents. On the other hand, multi-agents based routing protocols have been proposed [8,11,12]. Ant-like agents behave not only as explorers for the route finding, but also as messengers to bring actual message data. They actively update the routing tables in the nodes as they determine the next nodes to visit. Agents can react autonomously in the face of changing circumstances. They are more adaptable, and therefore they are more suitable for MANETs.

Minar *et al.* proposed a dynamic routing protocol using mobile agents [8]. Each node has a routing table that stores route information for every destination. As in AntNet and AntHocNet, ant-like agents, called *routing agents* are dispatched to explore the network. The node that a *routing agent* visits next is selected at the equivalent probability. The *routing agent* brings its own history of movement and updates the routing table of the node it is visiting. When a predefined wandering time has expired, then the *routing agent* disappears. When a source node wants to send a message to its destination, the node dispatches a series of *message agents* that are corresponding data packets. *Message agents* also determine their next hop autonomously by looking up the routing table at each node. They repeat these looking-up and hopping actions until they reach their destinations. When they reach their destination, they disappear.

Onishi *et al.* improved this model so that the *routing agent* also brings the time required to travel from the destination to the current node and the number of hops [47]. This history is also stored on the routing table. Kawarazaki *et al.* further extended this model so that the network does not lose its effectiveness when the number of nodes changes [48]. Nishimura *et al.* still further extended this model so that the network can keep its effectiveness in the face of congestion [49]. The effectiveness is achieved by the *routing agents*. They collect and distribute information about network congestion and update the routing table at each node they visit. *Message agents* move from one node to another node based on this information. The *message agent* has an evaluation function to select a best route.

5. Discussions and Conclusions

We have reviewed different agent based routing protocols used in MANETs. Any routing protocols used in MANET share common shortcomings. The methods based on the reactive ant-like agents have to wait until the communication routes from the source to the destination have been established [41–43]. For example, DSR and AODV, like the reactive ant-like agent approaches, take some time to discover the route, which delays the establishment of the connections in a large scale network [7]. On the other hand, with the methods based on the proactive ant-like agents, since the route from a source node to the destination may have been already found, the source node can transmit data packets without the delay.

The methods based on the proactive ant-like agents, however, may not function well because the network topology of a MANET is dynamic and lives of the routes are rather short [8,9,25–27]. They also have too much overhead for relatively the small bandwidth used in MANETs. In order to mitigate the overhead of *backward ants*, Schoonderwoed has proposed a routing algorithm that uses only *forward ants* to update the routing table of intermediate nodes [28,29]. The *forward ants* of this algorithm keep the path length from the source node, and update the routing table based on the information. The pheromone value at each intermediate node is given by the *forward ant* that comes through the shortest path from the source node. Whenever a node needs to send data packets to the source node, it can use the shortest path for the connection.

In traditional communication scenarios, the source node and destination node act symmetrically. Therefore, when a network uses only *forward ants*, other *forward ants* may go back to the destination to the source node resulting in the same situation using the *forward ants* and *backward ants*. In multi-agent systems, however, the communication is asymmetrical. A message, or even a complete task, is conveyed by an agent. Therefore, using only *forward ants* to explore the network is then a good idea. In addition, unlike in AntNet or AntHocNet where the nodes have learning ability to cope with the changing network topology (by sending *hello messages* to each other), in the multi-agent environment, mobile agents keep the information about the network and learn as they move onto the network so that the *backward ants* that update the routing tables are not necessary. Mobile agents utilize the routing tables as shared data structures by which they exchange information with each other. Routing tables are updated by the mobile agents only for exchanging routing information between mobile agents.

The author has conducted resource discovery algorithms based on the ant colony optimization in peer-to-peer networks [50,51]. The idea that ant-like agents roam in the network to look for a particular resource has some similarities to finding topology of a network. We have recently expanded the idea to not only data traffic but also human refugees' flow [52]. Each node, *i.e.*, human refugee, has a smartphone with GPS so that the smartphone senses the flow of the people and guides him to the most promising evacuation route. In such situations, we cannot rely on a base station, and have to depend on *ad hoc* networks. The *ad hoc* network has many promising applications. We are witnessing the coming era of ubiquitous computing through MANETs, and the more the applications that are developed the more efficient routing protocols are required. The Ant-AODV like hybrid routing protocol may provide high connectivity with minimum overhead for processing ant-like agents.

In multi-agent environments, not only route exploration ants but also data packets, or even complete tasks may be incorporated into mobile agents. Therefore establishing a communication route from the

source node to the destination node before dispatching mobile agents is not necessary and it is natural not to use backward ants to update routing tables to establish the best route. In such a model, each agent has to have some kind of evaluation function to select a best route at each intermediate node. As Nishimura points out, however, it is almost impossible to determine a generic function [49]. There are too many factors that affect such a function: for example, the positional relationship among nodes, the change over time of frequency of sending data packets, and the change of network topology. The evaluation function may be tailored for a particular environment. Incorporating learning ability into mobile agents is a possible research direction we can pursue.

In this paper, we have focused the domain of the ant colony optimization framework for routing protocols, because we believe that exploring those protocols is beneficial for constructing systems that employ multiple mobile agents in which we are currently engaged. A more general encyclopedically comprehensive survey on the routing protocols in MANETs can be found in [53].

For the next step of the review about the routing protocols based on ant-like mobile agents we would like to pursue the different aspects of those protocols such as route congestions and the broadcast behaviors in network performance. Those issues naturally denote the future directions.

Acknowledgments

Kimiko Gosney has provided useful comments and the author appreciates them. He also thanks anonymous reviewers who gave him many valuable comments.

Conflict of Interest

The authors declare no conflict of interest.

References

1. Perkins, C. *Ad Hoc Networking*; Addison-Wesley Publishers: Boston, MA, USA, 2001.
2. Portmann, M.; Pirzada, A.A. Wireless mesh networks for public safety and crisis management applications. *IEEE Int. Comput.* **2008**, *12*, 18–25.
3. Qiangfeng, J.; Manivannan, D. Routing Protocol for Sensor Networks. In Proceedings of IEEE Consumer Communications and Networking Conference, Las Vegas, NV, USA, 5–8 January 2004; pp. 93–98.
4. Vladimirova, T.; Xiaofeng, W.; Bridges, C.P. Development of a Satellite Sensor Network for Future Space Missions. In Proceedings of IEEE Aerospace Conference, Big Sky, MT, USA, 1–8 March 2008; pp. 1–10.
5. Perkins, C.E.; Bhagwat, P. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. *ACM SIGCOMM Comput. Commun. Rev.* **1994**, *24*, 234–244.
6. Johnson, D.B.; Maltz, D.A.; Broch, J. DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless *Ad Hoc* Networks. In *Ad Hoc Networking*; Perkins, C.E., Ed.; Addison-Wesley: Boston, MA, USA, 2001; pp. 139–172.
7. Perkins, C.; Belding-Royer, E.; Das, S. *Ad Hoc On-Demand Distance Vector (AODV) Routing*. In Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, USA, 25–26 February 1999; pp. 90–100.

8. Minar, N.; Kramer, K.H.; Maes, P. Cooperating Mobile Agents for Dynamic Network Routing. In *Software Agents for Future Communication Systems*; Hayzelden, A., Ed.; Springer-Verlag: Berlin Heidelberg, Germany, 1999; pp. 287–304.
9. Choudhury, R.R.; Bandyopadhyay, S.; Paul, K. A Distributed Mechanism for Topology Discovery in *Ad Hoc* Wireless Networks Using Mobile-Agents. In Proceedings of the 1st ACM International Symposium on Mobile *Ad Hoc* Networking & Computing, Boston, MA, USA, 11 August 2000; pp. 145–146.
10. Carrillo, L.; Marzo, J.L.; Vilà P.; Mantilla, C.A. MAntS-Hoc: A Multi-agent Ant-based System for Routing in Mobile *Ad Hoc* Networks. In Proceedings of the Catalan Congress of Artificial Intelligence, Barcelona, Spain, October 2004; pp. 285–292.
11. Amin, K.A.; Mikler, A.R. Dynamic Agent Population in Agent-Based Distance Vector Routing. In Proceedings of Second International Workshop on Intelligent Systems Design and Applications, Atlanta, GA, USA, 7–8 August 2002.
12. Amin, K.A.; Mikler, A.R. *Towards Resource Efficient and Scalable Routing: An Agent-based Approach*, Technical Report, University of North Texas, Department of Computer Science, 2002.
13. Schoonderwoed, R.; Holland, O.; Bruten, J. Ant-Like Agents for Local Balancing in Telecommunications Networks. In Proceedings of the First International Conference on Autonomous Agents, Marina del Rey, CA, USA, 5–8 February 1997; pp. 209–216.
14. Bertsekas, D.; Gallager, R. *Data Networks*; Prentice-Hall: Upper Saddle River, NJ, USA, 1987.
15. Dorigo, M.; Gambardella, L.M. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Trans. Evol. Comput.* **1997**, *1*, 53–66.
16. Appleby, S.; Steward, S. Mobile software agents for control in telecommunications networks. *Br. Telecom Technol. J.* **2000**, *18*, 68–70.
17. Subramanian, D.; Druschel, P.; Chen, J. Ants and Reinforcement Learning: A Case Study in Routing in Dynamic Networks. In Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, Nagoya, Japan, 23–29 August 1997; pp. 832–839.
18. Bonabeau, E.; Henaux, F.; Guéin, S.; Snyers, D.; Kuntz, P.; Theraulaz, G. Routing in Telecommunications Networks with Ant-Like Agents. In Proceedings of the Second International Workshop on Intelligent Agents for Telecommunication Applications, Paris, France, 4–7 July, 1998; Lecture Notes in Computer Science; Springer-Verlag: London, UK, 1998; Volume 1437, pp. 60–71.
19. Deneubourg, J.L.; Goss, S. Collective patterns and decision-making. *Ethol. Ecol. Evol.* **1989**, *1*, 295–311.
20. Beckers, R.; Deneubourg, J.L.; Goss, S.; Pasteels, J.M. Collective decision making through food recruitment. *Insectes Sociaux* **1990**, *37*, 258–267.
21. Coloni, A.; Dorigo, M.; Maniezzo, V. Distributed Optimization by Ant Colonies. In Proceedings of the European Conference on Artificial Life, Paris, France, 1–3 December 1991; pp. 134–142.
22. Dorigo, M.; Gambardella, L.M. Ant colony system: A cooperative learning approach to the traveling salesman. *IEEE Trans. Evol. Comput.* **1997**, *1*, 53–66.
23. Dorigo, M.; Maniezzo, V.; Coloni, A. *Positive Feedback as a Search Strategy*; Technical Report 91–016; Dipartimento di Elettronica, Politecnico di Milano, June 1991.
24. Dorigo, M.; Maniezzo, V.; Coloni, A. The ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. Part B Cybernetics* **1996**, *26*, 29–41.

25. Di Caro, G.; Dorigo, M. *AntNet: A Mobile Agent Approach to Adaptive Routing*; Technical Report IRIDIA 97–12; Université Libre de Bruxelles, Brussels, Belgium, December 1997.
26. Di Caro, G.; Dorigo, M. An Adaptive Multi-agent Routing Algorithm Inspired by Ants Behaviour. In Proceedings of Fifth Annual Australasian Conference on Parallel and Real-Time Systems, Adelaide, Australia, 28–29 September 1998.
27. Di Caro, G.; Dorigo, M. AntNet: Distributed stigmergetic control for communications networks. *J. Artif. Intell. Res.* **1998**, *9*, 317–365.
28. Schoonderwoerd, R.; Holland, O.; Bruten, J.; Rothkrantz, L. Ant-based load balancing in telecommunications networks. *Adapt. Behav.* **1996**, *5*, 169–207.
29. Schoonderwoerd, R.; Holland, O.; Bruten, J. Ant-like Agents for Load Balancing in Telecommunications Networks. In Proceedings of the First International Conference on Autonomous Agents, Marina del Rey, CA, USA, 5–8 February 1997; pp. 209–216.
30. Di Caro, G.; Dorigo, M. Two Ant Colony Algorithms for Best-Effort Routing in Datagram Networks. In Proceedings of the Tenth International Conference on Parallel and Distributed Computing and Systems, Las Vegas, NV, USA, 28–31 October 1998.
31. Di Caro, G.; Ducatelle, F.; Gambardella, L.M. *AntHocNet: An Adaptive Nature-Inspired Algorithm for Routing in Mobile Ad Hoc Networks*; Technical Report IDSIA-27-04-2004; Dalle Molle Institute for Artificial Intelligence, Manno, Switzerland, September 2004.
32. Di Caro, G.; Ducatelle, F.; Gambardella, L.M. AntHocNet: An Ant-Based Hybrid Routing Algorithm for Mobile Ad Hoc Networks. In Proceedings of Eighth International Conference on Parallel Problem Solving from Nature VIII, Birmingham, UK, September 2004; Lecture Notes in Computer Science, Volume 3242, pp. 461–470.
33. Ducatelle, F.; di Caro, G.; Gambardella, L.M. Ant Agents for Hybrid Multipath Routing in Mobile Ad Hoc Networks. In Proceedings of the Second Annual Conference on Wireless On-demand Network Systems and Services, St. Moritz, Switzerland, 19–21 January 2005; pp. 18–19.
34. Di Caro, G.; Ducatelle, F.; Gambardella, L.M. AntHocNet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks. *Eur. Trans. Telecommun.* **2005**, *16*, 443–455.
35. Mirabedini, S.J.; Teshnehlab, M.; Rahmani, A.M. FLAR: An adaptive, fuzzy routing algorithm for communications networks using mobile ants. *Cybernet. Syst.* **2008**, *39*, 684–702.
36. Wankhade, S.B.; Ali, M.S. Route failure management technique for ant based routing in MANET. *Int. J. Sci. Eng. Res.* **2011**, *2*, 1–5.
37. Bouazizi, I. ARA—The Ant-Colony Based Routing Algorithm for MANETs. In Proceedings of the 2002 International Conference on Parallel Processing Workshops, Vancouver, BC, Canada, 21 August 2002; pp. 79–85.
38. Roth, M.; Wicker, S. Asymptotic Pheromone Behavior in Swarm Intelligent MANETs. In Proceedings of the Conference on Mobile and Wireless Communication Networks, Paris, France, 25–27 October 2004; pp. 335–346.
39. Gunes, M.; Kahmer, M.; Bouazizi, I. Ant Routing Algorithm (ARA) for Mobile Multi-hop Ad Hoc Networks New Features and Results. In Proceedings of the 2nd Mediterranean Workshop on Ad-Hoc Networks (Med-Hoc-Net'2003), Mahdia, Tunisia, 25–27 June 2003; pp. 25–27.

40. Arif, M.; Rani, T. Enhanced Ant Colony based Routing in MANETs. In Proceedings of 5th IEEE International Conference on Advanced Computing & Communication Technologies, Panipat, India, 5 November 2011; pp. 48–54.
41. Cauvery, N.K.; Viswanatha, K.V. Enhanced ant colony based algorithm for routing in mobile ad hoc network. *World Acad. Sci. Eng. Technol.* **2008**, *22*, 30–35.
42. Woo, M.; Dung, N.H.; Roh, W.J. An Efficient Ant-based Routing Algorithm for MANETs. In Proceedings of the 10th International Conference on Advanced Communication Technology, Phoenix Park, South Korea, 17–20 February 2008; pp. 933–937.
43. Zhou, Y.; Zincir-Heywood, A.N. Intelligent Agents for Routing on Mobile Ad-Hoc Networks. In Proceedings of the Second Annual Conference on Communication Networks and Services Research, May 2004; pp. 249–254.
44. Baras, J.S.; Mehta, H. A Probabilistic Emergent Routing Algorithm (PERA) for Mobile Ad Hoc Networks. In Proceedings of the WiOpt '03: Modeling and Optimization in Mobile, AdHoc and Wireless Networks, Sophia-Antipolis, France, 3–5 March 2003.
45. Câmara, D.; Loureiro, A.A.F. GPS/ant-like routing in Ad Hoc networks. *Telecommun. Syst.* **2001**, *18*, 85–100.
46. Marwaha, S.; Tham, C.K.; Srinivasan, D. A Novel Routing Protocol Using Mobile Agents and Reactive Route Discovery for Ad Hoc Wireless Networks. In Proceedings of the 10th IEEE International Conference on Networks, Paris, France, 12–15 November 2002; pp. 311–316.
47. Onishi, R.; Yamaguchi, S.; Morino, H.; Aida, H.; Saito, T. A multi-agent system for dynamic network routing. *IEICE Trans. Commun.* **2001**, *E84-B*, pp. 2721–2728.
48. Kawarazaki, K.; Takahashi, K. An extension of the dynamic network routing system using mobile agents (in Japanese). *IEICE Trans. Inf. Syst.* **2003**, *J86-B*, pp. 2433–2436.
49. Nishimura, K.; Takahashi, K. A Multi-Agent Routing Protocol with Congestion Control for MANET. In Proceedings of the 21st European Conference on Modeling and Simulation, Prague, Czech, 4–6 June 2007; CD-ROM.
50. Kambayashi, Y.; Harada, Y. Integrating Ant Colony Optimization in a Mobile-Agent Based Resource Discovery Algorithm. In Proceedings of the IADIS International Conference Intelligent Systems and Agents, Algarve, Portugal, 21–23 June 2009; pp. 149–158.
51. Kambayashi, Y.; Harada, Y. A Resource Discovery Method Based on Multi-Agents in P2P Systems. In *Intelligent Agents in the Evolution of Web and Applications*; Nguyen, N.T., Jain, L.C., Eds.; Springer-Verlag: Berlin Heidelberg, Germany, 2009; pp. 113–135.
52. Aviles del Moral, A.; Takimoto, M.; Kambayashi, Y. ERAM: Evacuation Routing Using Ant Colony Optimization over Mobile Ad Hoc Networks. In Proceedings of the International Conference on Agents and Artificial Intelligence, Barcelona, Spain, 15–18 February 2013; pp. 118–127.
53. Boukerche, A.; Turgut, B.; Aydin, N.; Ahmad, M.Z.; Bölöni, L.; Turgut, D. Routing protocols in ad hoc networks: A survey. *Comput. Netw.* **2011**, *55*, 3032–3080.