A Taxonomy of Routing Protocols in Ad hoc Networks

Azzedine Boukerche, Mohammad Z. Ahmad, Begumhan Turgut, and Damla Turgut*

1 Introduction

Ad hoc networks are wireless networks without a fixed infrastructure, which are usually assembled on a temporary basis to serve a specific deployment such as emergency rescue or battlefield communication. They are especially suitable for scenarios where the deployment of an infrastructure is either not feasible or is not cost effective. The differentiating feature of an ad hoc network is that the functionality normally assigned to infrastructure components, such as access points, switches and routers needs to be achieved by the regular nodes participating in the network. For most cases, there is an assumption that the participating nodes are mobile, do not have a guaranteed uptime, and have limited energy resources.

In infrastructure based wireless networks, such as cellular networks or WiFi, the wireless connection goes only one-hop to the access point or the base station; the remainder of the routing happens in the wired domain. At most, the decision which needs to be made is which base station should a mobile node talk to, or how it should handle the transfer from one station to another during movement. Routing in the wired domain was long considered a mature field, where trusted and reliable solutions exist. The topology of the infrastructure, its bandwidth, routing and switching resources are provisioned to provide a good fit with the expected traffic.

In ad hoc networks, however, routing becomes a significant concern, because it needs to be handled by ordinary nodes which do not have neither specialized equipment, neither a fixed, privileged position in the network. Thus, the introduction of ad hoc networks signalled a resurgent interest in routing through the challenges posed by the mobility of the nodes, their limited energy resources, their heterogeneity (which under some conditions can lead to asymmetric connections) and many other issues. These challenges were answered with a large number of routing algorithms, and ad hoc routing remains an active and dynamically

*A. Boukerche is with School of Information Technology and Engineering, University of Ottawa, Ottawa, Canada. M.Z. Ahmad and D. Turgut are with School of Electrical Engineering and Computer Science, University of Central Florida Orlando, Florida. B. Turgut is with Department of Computer Science, Rutgers University, Piscataway, New Jersey.

evolving research area. Ad hoc routing algorithms are serving as a source of ideas and techniques to related technologies such as wireless sensor networks and mesh networks.

In this chapter, we survey the field of wireless ad hoc routing. While we attempted to include most of the influential algorithms, our survey cannot be exhaustive: the number of proposed algorithms and variations exceeds one thousand. However, we strived to represent most of the research directions in ad hoc routing thus giving the reader an introduction to the issues, the challenges and opportunities offered by the field.

2 Ad hoc networks applications

In this section, we present some applications of ad hoc networks [1]:

Conferencing: Mobile conferencing is without a doubt one of the most recognized applications. Establishment of an ad hoc network is essential for mobile users where they need to collaborate in a project outside the typical office environment.

Emergency Services: Responding to emergency situations such as disaster recovery is yet another naturally fitting application in the ad hoc networking domain. During the time of emergencies, several mobile users (policeman, firefighters, first response personnel) with different types of wireless devices need to not only communicate but also maintain the connectivity for long periods of time.

Home Networking: The wireless computers at home can also create an ad hoc network where each node can communicate with the others without taking their original point of attachment into consideration. This approach is alternative to assigning multiple IP addresses to each wireless device in order to be identified.

Embedded Computing Applications: Several ubiquitous computing [2] internetworking machines offer flexible and efficient ways of establishing communication methods with the help of ad hoc networking. Many of the mobile devices already have add-on inexpensive wireless components, such as PDAs with wireless ports and Bluetooth radio devices.

Sensor Dust: This application can be considered a combination of ad hoc and sensor networks. In hazardous or dangerous situations, it makes sense to distribute group of sensors with wireless transceivers to obtain critical information about the unknown site by the creation of ad hoc networks of these sensors.

Automotive/PC Interaction: The interaction between many wireless devices (laptop, PDA, and so on) being used in the car for different purposes can create an ad hoc network in order to carry out tasks more efficiently. An example can be finding the best possible mechanic shop to fix a car problem in a new city on the way to a meeting.

Personal Area Networks and Bluetooth: A Personal Area Network (PAN) creates a network with many of the devices that are attached or carried by a single user. Even though the communication of devices within PAN does not concern mobility issues, the mobility becomes essential when different PANs need to interact with each other. Ad hoc networks provide flexible solutions for inter-PAN communications. For instance, Bluetooth provides a wireless technology built-in to many of the current PDAs and up to eight PDAs, called *piconet*, can exchange information.

3 Design Issues

The main reasoning behind the design of mobile ad hoc networks was to respond to the military needs for battlefield survivability [1]. The situations in the battlefield require soldiers to move from place to place without any constraints by wireline communications and communicate with each other without depending on any fixed infrastructure. Since it is almost impossible to have a fixed backbone network in certain territories such as desert, timely deployment of mobile nodes communicating via wireless medium becomes critical. Another deciding point is the consideration of the physics of electromagnetic propagation. The fact is that frequencies much higher than 100 MHz are limited by their propagation distance. Therefore, for a mobile host to communicate with another mobile host beyond its transmission range, multi-hop routing protocols become necessary. This means that messages are transferred from one host to another via other intermediate hosts.

Mobile ad hoc networks inherit all the issues/problems related to mobile computing and wireless networking and perhaps even more due to lack of infrastructure in these types of networks. There are several unique design challenges that need to be considered including deployment, coverage, connectivity, and so on.

Deployment: Deployment of such networks are accomplished dynamically on the fly and the lifetime of these networks are usually short-lived. The deployment of ad hoc networks simply eliminates the cost of laying cables and maintenance of an infrastructure. In order to have a partially functioning network immediately, an incremental deployment with minimal configuration is possible. The requirements vary between different types of deployments such as commercial, military, emergency-operations, and so on.

Coverage: Adequate coverage of the entire area in question has to be provided to enable effective communication within devices which may not be within direct transmission ranges of each other. In many cases, the coverage area is determined by the particular deployed application. For example, in the home networking example, some of the nodes can be fixed static entities while others are mobile.

Connectivity: Another unique feature of ad hoc networking is the lack of connectivity due to mobility. Because the nodes can move at all times, the connectivity graph is continually changing. If all these nodes lie within the transmission range of each other, the network is said to be *fully* connected which results in a complete graph [3]. Since this situation does not generally hold in practice, routing is needed between any two nodes which are connected directly. The only way to connect any two nodes farther apart is via the intermediary nodes between them, which create the need for *multi-hop* routing. Ad hoc networks cannot pre-compute a static routing table; rather, they must dynamically adjust routing based on the mobility of the nodes. Due to the mobility and dynamic topology changes, the protocols are designed to keep the network structure *stable* as long as possible.

4 Ad hoc networks routing protocols

There have been many existing routing protocols for ad hoc networks emphasizing on different implementation scenarios. However, the basic goals have always been to devise a routing protocol which minimizes control overhead, packet loss ratio, and energy usage while maximizing the throughput. As these types of networks can be used in a variety of situations (disaster recovery, battlefields, conferences, and so on), they differ in terms of their requirements and complexities. The routing protocols in ad hoc networks can hence be divided into five categories based on their underlying architectural framework as follows and are shown in Figure 1.

- Source-initiated (Reactive or on-demand) Section 4.1

- Table-driven (Proactive) Section 4.2
- Hybrid Section 4.3
- Location-aware (Geographical) Section 4.4
- Multipath Section 4.5

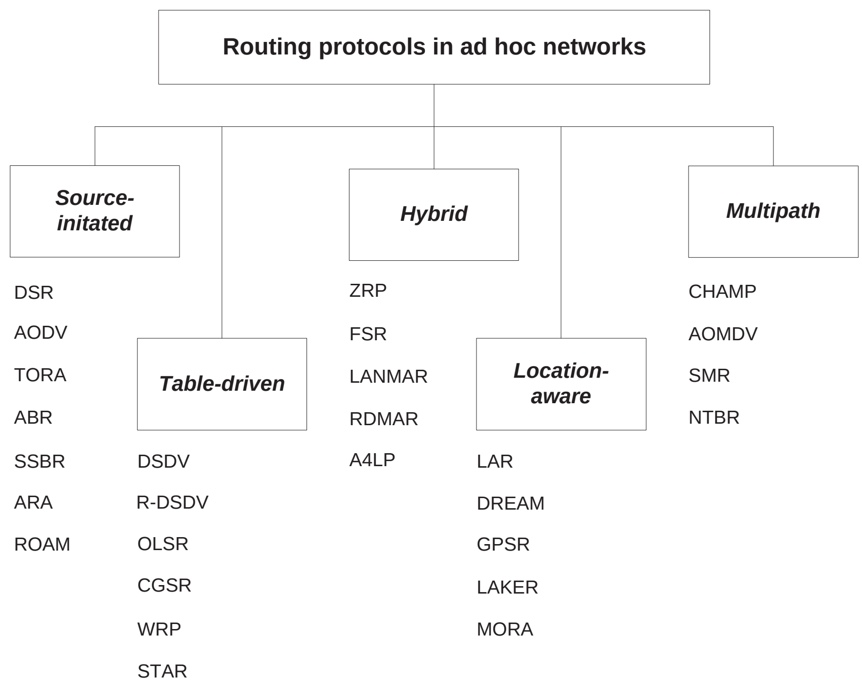


Figure 1: Categories of Ad hoc Routing Protocols

4.1 Source-initiated protocols

Source-initiated routing represents a class of routing protocols where the route is created only when the source requests a route to a destination. A route discovery procedure is invoked when the route is requested by the source and special route request packets are flooded to the network starting with the immediate neighbors. Once a route is formed or multiple routes are obtained to the destination, the route discovery process comes to an end. Route maintenance procedure maintains the active routes for the duration of their lifetimes.

Dynamic Source Routing (DSR) [4]: One of the most widely referred routing algorithms is Dynamic Source Routing (DSR) which is an "on-demand" routing algorithm and it has *route discovery* and *route* *maintenance* phases.

Route discovery contains both *route request* message and *route reply* messages. In route discovery phase, when a node wishes to send a message, it first broadcasts a route request packet to its neighbors. Every node within a broadcast range adds their node id to the route request packet and rebroadcasts. Eventually, one of the broadcast messages will reach either to the destination or to a node which has a recent route to the destination. Since each node maintains a *route cache*, it first checks its cache for a route that matches the requested destination. Maintaining a route cache in every node reduces the overhead generated by a route discovery phase. If a route is found in the route cache, then the node will return a route reply message to the source node rather than forwarding the route request message further. The first packet that reaches the destination node will have a complete route. DSR assumes that the path obtained is the shortest since it takes into consideration the first packet to arrive at the destination node. A route reply packet is sent to the source which contains the complete route from source to destination. Thus, the source node knows its route to the destination node and can initiate the routing of the data packets. The source caches this route in its route cache.

In route maintenance phase, two types of packets are used, namely *route error* and *acknowledgements*. DSR ensures the validity of the existing routes based on the acknowledgements received from the neighboring nodes that data packets have been transmitted to the next hop. Acknowledgement packets also include *passive acknowledgements* as the node overhears the next hop neighbor is forwarding the packet along the route to the destination. A route error packet is generated when a node encounters a transmission problem which means that a node has failed to receive an acknowledgement. This route error packet is sent to the source in order to initiate a new route discovery phase. Nodes upon receiving the route error message remove the route entry that uses the broken link within their route caches.

Ad hoc On-Demand Distance Vector (AODV) [5]: AODV routing protocol is developed as an improvement to the Destination-Sequenced Distance-Vector (DSDV) routing algorithm [6]. The aim of AODV is to reduce the number of broadcast messages sent throughout the network by discovering routes on-demand instead of keeping a complete up-to-date route information.

A source node seeking to send a data packet to a destination node checks its route table to see if it has a valid route to the destination node. If a route exists, it simply forwards the packets to the next hop along the way to the destination. On the other hand, if there is no route in the table, the source node begins a *route discovery* process. It broadcasts a *route request* (RREQ) packet to its immediate neighbors

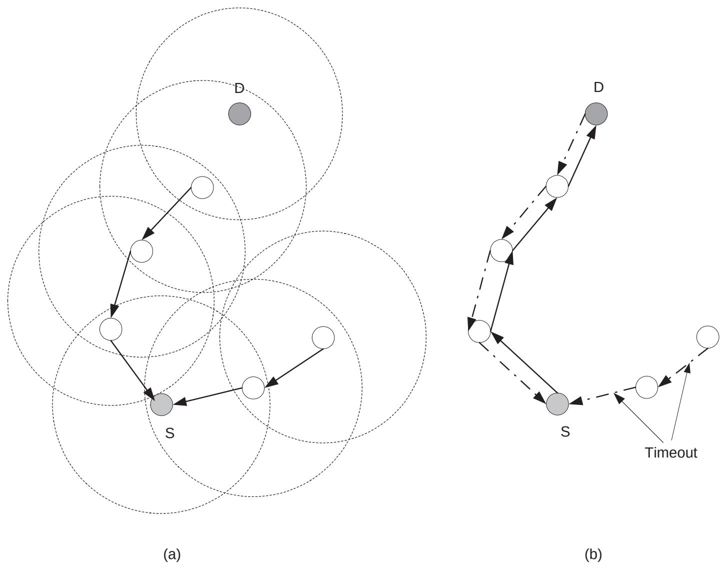


Figure 2: (a)Reverse path formation (b)Forward path formation [5].

and those nodes broadcast further to their neighbors until the request either reaches an intermediate node with a route to the destination or the destination node itself. This route request packet contains the IP address of the source node, current sequence number, the IP address of the destination node and the sequence number known last. Figure 2 denotes the forward and reverse path formation in the AODV protocol. An intermediate node can reply to the route request packet only if they have a destination sequence number that is greater than or equal to the number contained in the route request packet header. When the intermediate nodes forward route request packets to their neighbors, they record in their route tables the address of the neighbor from which the first copy of the packet has come from. This recorded information is later used to construct the reverse path for the route reply (RREP) packet. If the same RREQ packets arrive later on, they are discarded. When the route reply packet arrives from the destination or the intermediate node, the nodes forward it along the established reverse path and store the forward route entry in their route table by the use of symmetric links. Route maintenance is required if either the source or the intermediate node moves away. If a source node becomes unreachable, it simply re-initiates the route discovery process. If an intermediate node moves, it sends a *link failure notification* message to each of its upstream neighbors to ensure the deletion of that particular part of the route. Once the message reaches to source node, it then re-initiates the route discovery process.

Local movements do not have global effects as it was the case in DSDV. The stale routes are discarded and as a result no additional route maintenance is required. AODV has a route aging mechanism; however, it does not find out how long a link might be alive for routing purposes. The latency is minimized due to avoidance of using multiple routes. Integration of multicast routing makes AODV different than other routing protocols. AODV combines unicast, multicast and broadcast communications; currently uses only symmetric links between neighboring nodes. AODV provides both a route table for unicast routes and a multicast route table for multicast routes. The route table stores the destination and next-hop IP addresses and destination sequence number. Destination sequence numbers are used to ensure that all routes are loop free and the most current route information is used whenever route discovery is executed. In multicast communications, each multicast group has its own sequence number that is maintained by the multicast group leader. AODV deletes invalid routes by the use of a special route error message called Route Error (RERR).

Algorithm 1 AODV routing protocol [5]

```
// S is the source node; D is the destination node
// RT = Routing Table

S wants to communicate with D if RT of S contains a route to D
   S establishes communication with D
else
   S creates a RREQ packet and broadcasts it to its neighbors
   // RREQ contains the destination Address(DestAddr),
   // Sequence Number (Seq) and Broadcast ID (BID)
   for all nodes N receiving RREQ
      if (RREQ was previously processed)
         discard duplicate RREQ
      end if
      if (N is D)
         send back a RREP packet to the node sending the RREQ
      else if (N has a route to D with SeqId >= RREQ.Seq)
         send back a RREP packet
      else
         record the node from which RREQ was received
         broadcast RREQ
      end if
   end for
   while (node N receives RREP) and (N != S)
      forward RREP on the reverse path
      store information about the node sending RREP in the RT
   end for
   S receives RREP
   S updates its RT based on the node sending the RREP
   S establishes communication with D
end if
```

Temporally Ordered Routing Algorithm (TORA) [7]: TORA is adaptive and scalable routing algorithm based on the concept of link reversal. It finds multiple routes from source to destination in a highly dynamic mobile networking environment. An important design concept of TORA is that control messages are localized to a small set of nodes nearby a topological change. Nodes maintain routing information about their immediate one-hop neighbors. The protocol has three basic functions: route creation, route maintenance, and route erasure.

Nodes use a "height" metric to establish a directed cyclic graph (DAG) rooted at the destination during the route creation and route maintenance phases. The link can be either an upstream or downstream based on the relative height metric of the adjacent nodes. TORA's metric contains five elements: the unique node ID, logical time of a link failure, the unique ID of a node that defined the new reference level, a reflection indicator bit, and a propagation ordering parameter. Establishment of DAG resembles the query/reply process discussed in Lightweight Mobile Routing (LMR) [8]. Route maintenance is necessary when any of the links in DAG is broken. Figure 3 denotes the control flow for the route maintenance in TORA.

The main strength of the protocol is the way it handles the link failures. TORA's reaction to link failures is *optimistic* that it reverses the links to re-position the DAG for searching an alternate path. Effectively, each link reversal sequence searches for alternative routes to the destination. This search mechanism generally requires a *single-pass* of the distributed algorithm since the routing tables are modified simultaneously during the outward phase of the search mechanism. Other routing algorithms such as LMR uses two-pass whereas both DSR and AODV use three-pass procedure. TORA achieves its single-pass procedure with the assumption that all the nodes have synchronized clocks (via GPS) to create a temporal order of topological change of events. The "height" metric is dependent on the logical time of a link failure.

Associativity-Based Routing (ABR) [10]: ABR uses the property of "associativity" to decide on which route to choose. In this algorithm, route stability is the most important factor in selecting a route. Routes are discovered by broadcasting a *broadcast query* request packet and with the assistance of these packets, the destination becomes aware of all possible routes between itself and the source. Based on these available routes, a path is selected using the associativity property of these routes.

The ABR algorithm maintains a "degree of associativity" by using a mechanism called *associativity ticks*. According to this, each node in the network maintains a tick value for each of its neighbors. Every periodic link layer HELLO message increases the tick value by one each time it is received from a neighbor. Once the tick value reaches a specified threshold value, it means that the route is *stable*. If the neighbor goes out

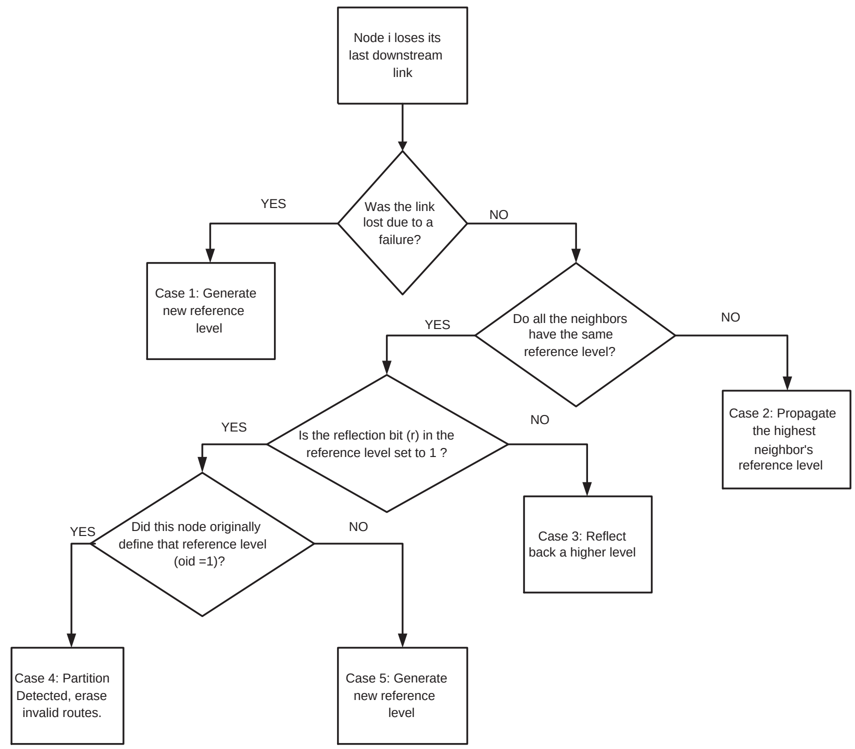


Figure 3: Flow diagram of route maintenance in TORA [9].

of the range, then the tick value is reset to zero. Hence a tick level above the threshold value is an indicator of a rather stable association between these two nodes.

Once a destination has received the *broadcast query* packets, it has to decide which path to select by checking the tick-associativity of the nodes. The route with the highest degree of associativity is selected since it is considered the most stable of the available routes.

ABR is quite an effective algorithm in selecting routes as it focuses on the route stability to a great extent. However some inherent drawbacks include memory requirements for the routing tables, excessive storage needs for storing the ticks, additional computation to maintain the tick count along with greater power requirements.

Signal Stability-Based Adaptive Routing (SSBR) [11]: The SSBR differs when compared with the conventional routing algorithms. The main routing criteria is the signal and location stability. The basic routing framework is similar to any other standard on-demand routing protocol: the route request is broadcast throughout the network, the destination replies back with the route reply message and then the sender sends data through the selected route. However, the signal strength (link quality) between neighboring nodes plays a major role in the route selection process in this protocol.

SSBR is comprised of two subprotocols: *Dynamic Routing Protocol (DRP)* and *Static Routing Protocol (SRP)*. The DRP interacts with the network interface device driver dealing with signal strengths through an API to determine the actual strength of a received signal. Using this signal information the DRP maintains a *signal stability table* which categorizes each link with the neighboring nodes as strong or weak. This table is updated with every new packet received. For instance, if a HELLO packet is received, the signal strength is monitored and the signal stability table is upgraded while for other packets such as route update packets, data packets, and so on, the packet is sent to the SRP for further processing. The SRP performs the routine tasks such as forwarding packets according to the existing routing table, replying to route requests, and so on.

The route request is given an option on the type of link it requests, i.e., strong, weak or a combination of both. If the route request specifies only strong links, all the route request packets coming from a perceived weaker link are dropped. Thus, the final discovered path consists of only strong links. If there are multiple paths from source to destination using strong links, the destination can choose among them. The destination can simply choose the first route request it receives. If however no strong links are found, the protocol could fall back on other available weaker links.

Two enhancements to the selection process are proposed. In the first case, the link strength (strong or link) is added for each hop into the route request packet and then forwarded towards the destination. In this case, the destination does not select the first route request packet received, but waits for a period of time to choose the best route among all the route requests within a set time interval. The second improvement suggests that any intermediate node can make an unnecessary route reply for a route it already has a prior information about.

The SSBR algorithm uses a similar scheme as the ABR algorithm to determine the reliability of links before selecting a particular route. While the former uses signal strength the latter uses ticks; however the goal of selecting a route with greatest reliability remains the same.

The Ant-colony based Routing Algorithm (ARA) [12]: This work presents a novel technique for ad hoc routing by using the concepts of *swarm intelligence* and the *ant colony* based meta heuristic. These ant colony algorithms focus on solving the complex problems by cooperation and without any direct communication. Ants communicate with each other using *stigmergy*, a means of indirect communication between individuals by modifying their operational environment.

Ants find the shortest path to a destination by making the use of *pheromones* 4. Initially, the ants randomly choose the path. On the way, they drop off pheromones. The ants chosen the shortest path reach the destination earlier than the others and travel back along the same path. They keep releasing additional pheromones on this path gradually increasing the pheromone concentration. The successive groups of ants coming from the source recognize the path with greater pheromone levels due to the higher concentration and hence start taking this route which ends up as the best route. This method of selecting the shortest path is used to determine the optimal route for ad hoc networks. This technique performs well with the dynamic link changes due to node mobility along with scalability issues.

Using the ant colony meta heuristic, the pheromone change of an edge $e(v_i, v_j)$ when moving from node v_i to v_j can be derived as follows: $\phi_{i,j} := \phi_{i,j} + \delta_\phi$. The pheromone concentration also decreases exponentially with time according to the following equation: $\phi_{i,j} := (1 - q).\phi_{i,j}$.

The algorithm has the following phases:

Route discovery. This phase uses two types of control packets: the *forward ant* (FANT) and the *backward ant* (BANT). The FANT establishes the pheromone track to the source node while the BANT establishes the pheromone track to the destination. When the route is required, the source broadcasts FANT packets to all its neighbors. FANTs are distinguished by unique sequence numbers. A node which receives a FANT for the first time creates a routing table record which contains the destination address, next hop, pheromone value. The source address of the FANT is taken as the destination address, the previous node address as the next hop, and the pheromone value is calculated based on the total number of hops required by the FANT to reach a particular node. When the FANT reaches the destination, the node updates its own information and sends the BANT back. Once, the BANT reaches the source, the path is to be used. Figure 5 and Figure 6 6 denote the forward and backward ants' route discovery phases.

Route maintenance. No special route maintenance packets are required by ARA as it uses the transmitted data packets to maintain the route. The pheromone value of the path is increased by δ_ϕ each time a data packet is sent along this path and it also decreases according to the equation above.

Route failure handling. Due to node mobility, route failures which are determined by missing acknowledgements may occur frequently. If a route failure occurs, an existing alternate path is used, otherwise, the source initiates the route discovery phase.

The expected overhead of ARA is very low since the route maintenance is carried out by the data packets.

Routing On-demand Acyclic Multipath (ROAM) [13]: The ROAM algorithm uses coordination among

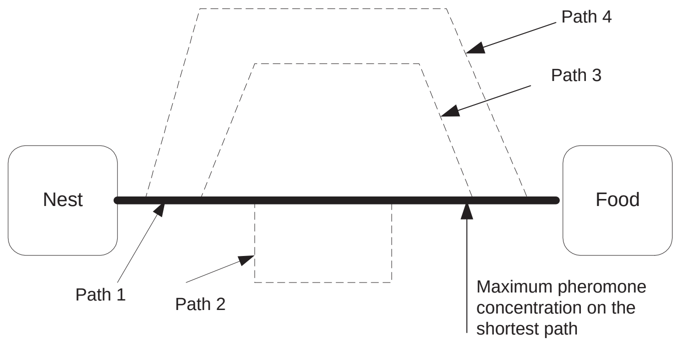


Figure 4: Pheromone concentration along the shortest path used by ants to discover food from their nest [12].

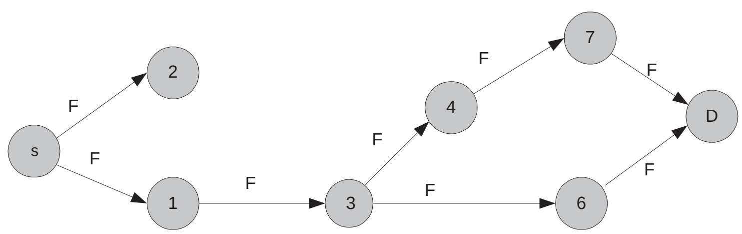


Figure 5: Forward ant route discovery phase. A forward ant (F) is send from the sender (S) toward the destination node (D). The forward ant is relayed by other nodes, which initialize their routing table and the pheromone values [12].

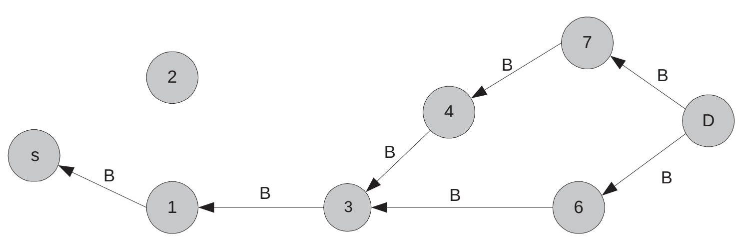


Figure 6: Backward aunt route discovery phase. The backward ant (B) has the same task as the forward ant. It is send by the destination node toward the source node [12].

nodes along directed acyclic subgraphs which are defined only on the routers' distances to the respective destinations. It is an extension of the DUAL [14] routing algorithm. The main motivation from the fact that conventional on-demand schemes tend to use flooding during route discovery repeatedly until a destination is obtained. If no route is found out initially, the source does not know whether to initiate another route discovery. This may be a problem when a malicious router indefinitely queries the network for a non-existence route causing network congestion. Standard protocols have no mechanisms to protect against such type of attacks. In ROAM, a search query either results in the destination path or all the routers determine that the destination is unreachable.

Each router in ROAM maintains *distance*, *routing*, and *link cost* tables. While the distance table maintains the distances of nodes for each destination and neighbors from the respective node, the routing table contains a column vector containing the distance to each destination, the feasible distance and the reported distance. The link cost table provides the link costs to each of the adjacent neighbors of the router. *Queries*, *replies*, and *updates* are the three types of control packets used in the routing protocol.

A router updates its routing table for a destination when it needs to: (i) add an entry for a particular destination; (ii) modify its distance to the destination; and (iii) erase the entry for the destination.

The routers in ROAM are either in *active* or *passive* states. If a router has send queries to all its neighbors and awaiting a reply, it is in active state, otherwise in passive state. Selection of loop-free paths allows a router select a neighbor as its successor only if its is a *feasible successor*. This provides a shortest loop-free path to the destination and is determined by two different algorithms based on the fact that they are either passive or active. A diffusing search starts by a router when it requests a path to a destination and this packet is propagated through routers which have no entry of the node. The first router which has an available route to the destination responds to the source with the distance to the node. At the end of this search, either the source has a finite distance to the destination or realizes that the destination is unreachable. Link costs are also updated based on the packets received.

The ROAM provides loop-free multipaths if the successors are selected using the passive and active successor algorithms. The very nature of this algorithm makes it suitable for wireless networks with limited mobility.

4.2 Table-driven protocols

Table driven protocols always maintain up-to-date information of routes from each node to every other node throughout the network. Routing information is stored in a routing table in each of the nodes and route updates are propagated throughout the network to keep the routing information as recent as possible. Different protocols keep track of different routing state information; however, each with the common goal of reducing route maintenance overhead as much as possible. These types of protocols are not suitable for highly dynamic networks due to the extra control overhead generated to keep the routing tables consistent and fresh for each node in the network.

Destination-Sequenced Distance-Vector (DSDV) [6]: The Destination-Sequenced Distance-Vector (DSDV) is a table-driven routing protocol based on Bellman-Ford routing algorithm. Every mobile node maintains a routing table which contains all of the possible destinations in the network and each individual hop counts to reach those destinations. Each entry also stores a sequence number which is assigned by the destination. Sequence numbers are used to identify stale entries and avoidance of loops. In order to maintain routing table consistency, routing updates are periodically sent throughout the network. Two types of updates can be employed; *full dump* and *incremental*. A *full dump* sends the entire routing table to the neighbors and can require multiple network protocol data units (NPDUs). *Incremental* updates are smaller that must fit in a packet and used to transmit those entries from the routing table since the last full dump update. When a network is stable, incremental updates are sent and full dump are usually infrequent. On the other hand, full dumps will be more frequent in a fast moving network. The mobile nodes maintain another routing table to contain the information sent in the incremental routing packets. In addition to the routing table information, each route update packet contains a distinct sequence number that is assigned by the transmitter. The route labeled with the most recent (highest number) sequence number is used. The shortest route is chosen if any of the two routes have the same sequence number.

Analysis of a Randomized Congestion Control Scheme with DSDV Routing in Ad hoc Wireless Networks [15]: In the randomized version of the DSDV (R-DSDV) protocol, the control messages are propagated based on a routing probability distribution instead of a periodic basis. Local nodes can tune their parameters to the traffic and route the traffic through other routes with lesser load. This implies implementing a congestion control schemen from the routing protocol's perspective.

The randomization of the algorithm is with respect to the routing table advertisement packets and the rate at which they are sent. In DSDV, whenever there is any change in the routing table, advertisement packets are propagated to update the state information at each node. R-DSDV sends these update messages only at a probability $Pr_{n,adv}$ for a node n in the network. This can reduce the control packet overhead; however, there may be a corresponding delay in updating all the nodes. If there is a routing table update at node n, the node can sends a regular message with a probability $1 - Pr_{n,adv}$ or an update message with probability $Pr_{n,adv}$. Thus, the rate at which routing table advertisement sent is $\rho_n = F_{send} x Pr_{n,adv}$, where F_{send} is the frequency at which a node is allowed to send a message. Piggybacking can be used with this scheme to transfer routing table updates with the regular messages.

Optimized Link State Routing (OLSR) [16]: The OLSR optimizes a pure link state since it reduces the size of information sent in each message and also reduces the total control overhead by minimizing the number of retransmissions flooding an entire network. It uses a multipoint relaying technique to flood the control messages in a network in an efficient manner.

The aim of using the multipoint relay is to reduce retransmissions within the same region. Each node selects a set of one-hop neighbors which are called the *multipoint relays (MPR)* for the node. The neighbors of the node which are not MPRs process the packets but do not forward them since only the MPRs forward the packets and the node forwards any of the broadcast messages to these MPR nodes.

The multipoint relay set must be chosen in an efficient manner to ensure that its range covers all the two-hops neighbors. This set must also be the minimum set to broadcast the least number of packets. The multipoint relay set of a node N should be such that every two-hops neighbor of N has a bi-directional link with the nodes in the MPR set of N. These bi-directional links can be determined by using periodic HELLO packets which contain information about all neighbors and their link status. Thus, a route is a sequence of hops from source to destination through multipoint relays within the network. Source does not know the complete routes only next hop information to forward the messages.

Clusterhead Gateway Switch Routing (CGSR) [17]: The CGSR protocol is a clustering scheme that uses a distributed algorithm called the Least Cluster Change (LCC). By aggregating nodes into clusters controlled by clusterheads, a framework for developing additional features for channel access, bandwidth allocation and routing is created. Nodes communicate with the clusterhead which in-turn communicate with other cluster heads within the network (see Figure 7).

Selecting a clusterhead is a very important task as frequently changing cluster heads will have an ad-

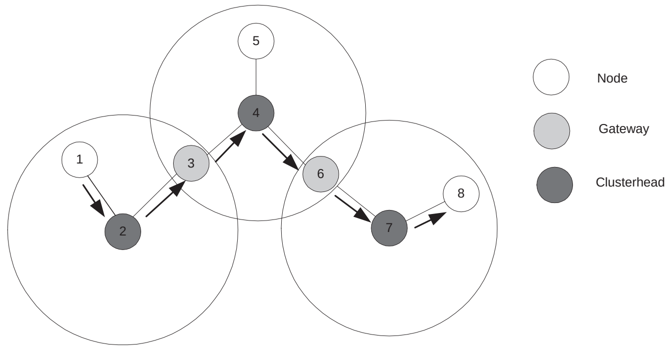


Figure 7: Cluster Gateway Switch Routing [17]

verse effect on the resource allocation algorithms which depend on it. Thus cluster stability is of primary importance in this scheme. The LCC algorithm is stable in which a clusterhead will change only under two conditions–when two clusterheads come within the range of each other or when a node gets disconnected from any other cluster.

CGSR is an effective way for channel allocation within different clusters by enhancing spatial reuse. The explicit requirement of CGSR on the link layer and MAC scheme are: each cluster is defined with unique CDMA code and hence each cluster is required to utilize spatial reuse of codes. Within each cluster, TDMA is used with token passing.

Gateway nodes are defined as those nodes which are members of more than one cluster and therefore need to be communicating using different CDMA codes based on their respective clusterheads. The main fators affecting routing in these networks are token passing (in clusterheads) and code scheduling (in gateways). This uses a sequence number scheme as in DSDV [6] to reduce stale routing table entries and gain loop-free routes. A packet is routed through a collection of these cluster heads and gateways in this protocol.

Wireless Routing Protocol (WRP) [18]: WRP is one of the earliest works on routing algorithms and is similar to the distributed Bellman-Ford algorithm. It is a table driven protocol where routing tables are maintained for all destinations. The routing table contains an entry for each destination with the next hop and a cost metric. The route is chosen by selecting a neighbor node that would minimize the path cost. Link costs are also defined and maintained in a separate table and various techniques are available to determine these link costs.

To maintain the routing tables, frequent routing update packets have to be sent. These are sent to all neighbors of a node and contain all the routes which the node is aware of. As the name suggests, these are

Algorithm 2 CGSR routing protocol [17]

```
// CMT = Cluster Member Table; RT = Routing Table

S wants to communicate with D S looks up the clusterhead of D in CMT
S looks up the next hop towards D in RT S forwards packet to
clusterhead of next hop while packet is not delivered
   if destination CH is within direct range of current CH
      current CH forwards packet to destination CH
      destination CH delivers packet to D
   else
      current CH forwards packets to Gateway Node
      gateway node sends packet to CH of next hop
   end if
end if
```

just update messages and hence only the recent path changes are sent in these messages and not the whole routing table. To keep the links updated, empty HELLO packets are sent at periodic intervals only if no other update messages need to be sent. These empty HELLO packets are not required to be acknowledged specifically.

Source-Tree Adaptive Routing (STAR) [19]: STAR is an efficient link state protocol. Each node maintains a source tree which contains preferred links to all possible destinations. Nearby soure trees exchange information to maintain up-to-date tables. A route selection algorithm is executed based on the propagated topology information to the neighbors. The routes are maintained in a routing table containing entries for the destination node and the next hop neighbor.

In this protocol, link state update (LSU) messages are used to update changes of the routes in the source trees. Since these packets do not time out, no periodic messages are required. STAR protocol provides two distinct approaches: optimum routing (ORA) and least overhead routing (LORA). The ORA approach obtains the shortest path to the destination while LORA minimizes the packet overhead. STAR also requires a neighbor protocol to make sure that each node is aware of its active neighbors. The STAR protocol has been further developed as SOAR [20].

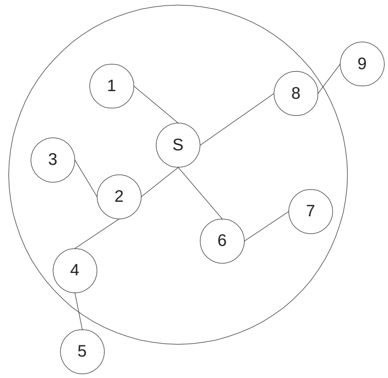


Figure 8: Example routing zone with $\rho = 2$.

4.3 Hybrid protocols

In hybrid routing schemes combine the power of on-demand and table-driven routing protocols. Static routing is generally used at the fringes of the network where route changes are not frequent while in the core of the network on-demand routing has more significance. These schemes create a bridge between the two major types of routing protocols and the overall performance obtained can be further improved.

Zone Routing Protocol (ZRP) [21]: ZRP is a well-known hybrid routing protocol and most suitable for large scale networks. Its name is derived from the use of "zones" which define the transmission radius for every participating node. This protocol uses a pro-active mechanism of node discovery within a node's immediate neighborhood while inter-zone communication is carried out by using reactive approaches.

ZRP utilizes the fact that node communication in ad hoc networks is mostly localized, thus the changes in the node topology within the vicinity of a node are of primary importance. ZRP makes use of this characteristic to define a framework for node communication with other existing protocols. Local neighborhoods, called *zones*, are defined for nodes. The size of a zone is based on ρ factor which is defined as the number of hops to the perimeter of the zone. There may be various overlapping zones which helps in route optimization. See Figure 8.

Neighbor discovery is accomplished by either Intrazone Routing Protocol (IARP) or simple "Hello" packets. IARP is pro-active approach and always maintains up-to-date routing tables. Since the scope of IARP is restricted within a zone, it is also referred to as "limited scope pro-active routing protocol". Route queries outside the zone are propagated by the route requests based on the perimeter of the zone (i.e. those with hop counts equal to ρ), instead of flooding the network.

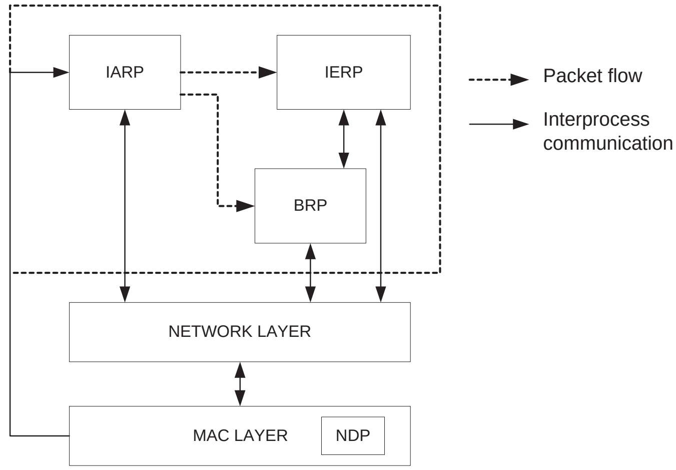


Figure 9: ZRP architecture [21].

The Interzone Routing Protocol (IERP) uses a reactive approach for communicating with nodes in different zones. Route queries are sent to peripheral nodes using the bordercast resolution protocol (BRP). Since a node does not resend the query to the node in which it received the query originally, the control overhead is significantly reduced and redundant queries are also minimized.

ZRP provides a hybrid framework of protocols, which enables a use of any routing strategy according to various situations. It can be optimized to take full advantage of the strengths of any current protocols. The ZRP architecture can be seen in Figure 9.

Fisheye State Routing (FSR) [22]: FSR is a hierarchical routing protocol which aims at reducing control packet overhead by introducing the multi-level scopes. It is essentially a table-driven protocol which implements the “fisheye” technique proposed in [23]. This technique is very effective to reduce the size of information required to represent graphical data. It uses the concept that the eye of a fish captures with greater detail the view nearer to the focal point while detail decreases as the distance from the focal point increases.

FSR is similar to link state routing as it maintains a routing table at each node. The only difference is in the maintenance of these tables. FSR introduces *scopes* concept which depends on the number of hops a packet traveled from its source. A higher frequency of update packets are generated for nodes within smaller scope while for farther away nodes, updates are fewer in general. Each node maintains a local topology map of the shortest paths which is exchanged periodically between the nodes.

Fisheye state routing allows distinct exchange periods for different entries in the routing tables. These scopes are considered based on the distance between each node. The foremost benefit is the reduction of the message size since the routing information of the far away nodes are omitted. With an increase in size of the network, a “graded” frequency update plan can be adopted across scopes to minimize the overall overhead. This protocols scales well to large size of networks while keeping the control overhead low without compromising on the accuracy of route calculations. Routes to farther destinations may seem stale; however, they become increasingly accurate as a packet approaches its destination.

Landmark Ad hoc Routing (LANMAR) [24]: This protocol combines properties of link state and distance vector algorithms and builds subnets of groups of nodes which are likely to move together. A *landmark* node is elected in each subnet, similar to FSR [22]. The key difference between FSR protocol is that LANMAR routing table consist of only the nodes within the scope and landmark nodes whereas FSR contains the entire nodes in the network its table. During the packet forwarding process, the destination is checked to see if it is within the forwarding node’s neighbor scope. If so, the packet is directly forwarded to the address obtained from the routing table. On the other hand, if the packet’s destination node is much farther, the packet is first routed to its nearest landmark node. As the packet gets closer to its destination, it acquires more accurate routing information, thus in some cases it may bypass the landmark node and routed directly to its destination. The link state update process is again similar to the FSR protocol. Nodes exchange topology updates with their one-hop neighbors. A distance vector, which is calculated based on the number of landmarks, is added to each update packet. As a result of this process, the routing tables entries with smaller sequence numbers are replaced with larger ones.

Relative Distance Micro-discovery Ad hoc Routing (RDMAR) [25]: The RDMAR protocol is very similar to existing reactive protocols since it uses the two standard phases of route-discovery and route-maintenance. However route discovery broadcast messages are limited by a maximum number of hops which is calculated using the relative distance between the source and destination. Each node also maintains a routing table which contains the next hop neighbor of each known destination, an estimated relative distance between all known source and destination nodes, a time stamp at which the current entry was made, a timeout field indicating the time at which a particular route is no longer active and a flag specifying if a route still exists or not.

The estimated distances are measured by the source nodes using the last known distance between the respective nodes, the last time when the route was updated and also the estimated speed of the destination node. Each node also maintains two other data structures–a *data retransmission buffer* which queues data being transmitted until an explicit acknowledgment is received and a *route request table* which stores all necessary information which pertains to the most recent route discovery.

Route discovery and route maintenance is carried out by broadcasting route request packets and expecting a route reply packet from the destination. Each node also occasionally probes for bi-directional links by sending a packet on the link where it has just received a packet. Route maintenance is performed when a route failure occurs and the node re-sends the data up to a maximum number of retries. This is why the intermediate nodes buffer data packets until they receive link level acknowledgments from the next-hop node. When a link failure occurs at an intermediate node close to the destination, this node sets the "emergency" flag in its route request packets such that it increases the possibility of a faster recovery time. If, however, the route has completely failed, the intermediate node forwards a *failure notification* to the source node by unicasting it to all neighboring nodes involved. When a node receives a failure notification, it updates its routing tables accordingly.

Scalable Location Update based Routing Protocol (SLURP) [26]: The SLURP focuses on developing an architecture scalable to large size networks. A location update mechanism maintains location information of the nodes in a decentralized fashion by mapping node IDs to specific geographic sub-regions of the network where any node located in this region is responsible for storing the current location information for all the nodes situated within that region. When a sender wishes to send a packet to a destination, it first queries nodes in the same geographic sub-region of the destination to get a rough estimate of its position. It then uses a simple geographic routing protocol to send the data packets. Since the location update cost is dependent on the speed of the nodes, for high speeds, more number of location update messages are generated. By theoretical analysis, it is shown that the routing overhead scales as $O(v)$ where v is the average node speed, and $O(N^{3/2})$ where N is the number of nodes within the network. It can be noted that the routing packet overhead scales linearly with respect to node speeds and with $N^{3/2}$ with the present number of nodes within the network.

A^4LP **routing protocol** [27, 28]: A^4LP is specifically designed to work in networks with asymmetric links. The routes to In-, Out-, and In/Out-bound neighbors are maintained by periodic neighbor update and immediately available upon request, while the routes to other nodes in the network are obtained by a path discovery protocol. A^4LP proposes an advanced flooding technique - *m-limited forwarding*. Receivers can re-broadcast a packet only if it qualifies a certain *fitness* value specified by the sender. The flooding cost is reduced and shortest high quality path is likely to be selected by using m-limited forwarding. Moreover, the metrics used to choose from multiple paths are based on the *power consumed per packet* and *transmission latency*. A^4LP, is also both *location-* and *power-aware* routing protocol supporting asymmetric links that may be suitable for heterogeneous MANET.

4.4 Location-aware protocols

Location-aware routing schemes in mobile ad hoc networks assume that the individual nodes are aware of the locations of all the nodes within the network. The best and easiest technique is the use of the Global Positioning System (GPS) to determine exact coordinates of these nodes in any geographical location. This location information is then utilized by the routing protocol to determine the routes.

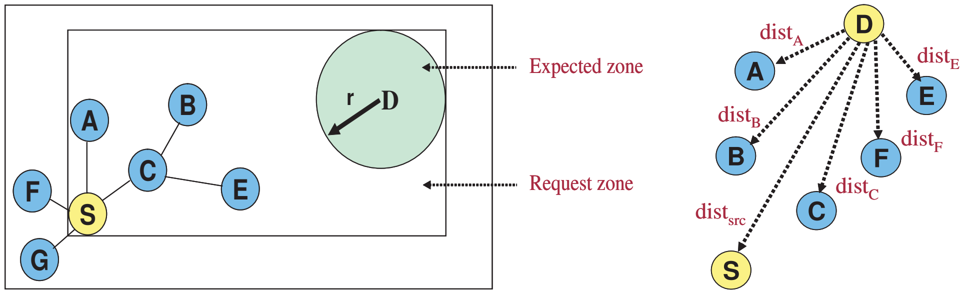
Location-Aided Routing (LAR) [29, 30]: The LAR protocol suggests an approach that utilizes location information to minimize the search space for route discovery towards the destination node. The aim of this protocol is to reduce the routing overhead for the route discovery and it uses the Global Positioning System (GPS) to obtain the location information of a node.

The intuition behind using location information to route packets is very simple and effective. Once the source node knows the location of the destination node and also has some information of its mobility characteristics such as the direction and speed of movement of the destination node, the source sends route requests to nodes only in the "expected zone" of the destination node. Since these route requests are flooded throughout the nodes in the expected zone only, the control packet overhead is considerably reduced. If the source node has no information about the speed and the direction of the destination node, the entire network is considered as the expected zone.

A source node before sending a packet determines the location of the destination node and defines its "request zone", the zone in which it initiates flooding with the route request packets. In some cases, the nodes outside the request zone may also be included. If the source node is not inside the destination node's expected zone, the request zone has to be increased to accommodate the source node. Also, a situation may occur where all neighboring nodes of the destination node may be located outside the request zone. In this case, the request zone has to be increased to include as many neighboring nodes as possible.

LAR defines two schemes to identify whether a node is within the request zone.

- *Scheme 1*: In this scheme, the source node simply includes the smallest rectangle containing the current



(a) (b)

Figure 10: LAR routing protocol. The diagrams (a) and (b) present LAR1 and LAR2 schemes [29, 30]

location of the source node and the expected zone of the destination node based on its initial location and current speed. The speed factor may be varied to either include the current speed or the maximum obtainable speed within the network. This expected zone will be a circle centered at the initial location of the destination node with a radius dependent on its speed of the movement. The source node sends the route request packets with the coordinates of the entire rectangle. The nodes receiving these packets check to see whether their own locations are within the zone. If so, they forward the packet using the regular flooding algorithm, otherwise the packets are simply dropped.

- *Scheme 2*: Here, the source node calculates the distance between itself and the destination node based on the GPS coordinates and includes these values within the route request packets. An intermediary node receiving this packet calculates its distance from the destination. If its distance from the destination is greater than that of the source, the intermediary node is not within the request zone and hence drops the packet. Otherwise, it forwards the packet to all its neighbors.

LAR essentially describes how location information such as GPS can be used to reduce the routing overhead in an ad hoc network and ensure maximum connectivity. See Figure 10

Distance Routing Effect Algorithm for Mobility (DREAM) [31]: The DREAM protocol also uses the node location information from GPS systems for communication. DREAM is a part proactive and part reactive protocol where the source node sends the data packet "in the direction" of the destination node by selective flooding. The difference from the other location based protocols is that only the data packets are forwarded to the next hop neighbor, not the control packets. Each node maintains table with the location information of each node and the periodic location updates are distributed among the nodes to keep this information as up-to-date as possible. Collectively updating location table entries indicates the proactive nature of the protocol while the fact that all intermediate nodes in a route perform a lookup and forward the data packet in the general direction of the destination, reflects DREAM's reactive properties.

DREAM is based on two classical observations: the *distance effect* and the *mobility effect*. The distance effect states that the greater the distance between two nodes, the slower they appear to move with respect to each other. Hence, the location information tables can be updated depending on the distance between the nodes without making any concessions on the routing accuracy. Two nodes situated farther apart view the other to be moving relatively slowly, requiring less frequent location updates compared with nodes closer to each other. The mobility effect determines how often the location information packets can be generated and forwarded. In an ideal scenario, whenever a node moves, it should update entire the network but not generate any packets if it remains idle. However, a node keeps generating location update packets at periodic intervals which can be a function of the node's mobility. Thus, the nodes with higher mobility generate more frequent location update messages. This allows each node to send control packets based on their mobility and helps to reduce the overhead by a great extent.

Since DREAM does not need any route discovery procedure, it does not incur the delay seen in other reactive protocols. It is energy and bandwidth efficient since control message generation is optimized with respect to node mobility. In addition, it is also inherently loop-free, robust and more importantly adaptive to mobility.

Algorithm 3 DREAM routing protocol: Send procedure [31]

```
// LT = Location Table

Find destination node from packet header D
if (no LT entry for D or the information is not valid)
    invoke recovery mode
else
    find existing neighbors from the LT
    if no neighbors exist
       invoke recovery mode
    else
       set the timer when the packet is forwarded
       transmit the packet to all the neighbors
    end if
end if
```

Greedy Perimeter Stateless Routing (GPSR) [32]: Similar to the other location based protocols,

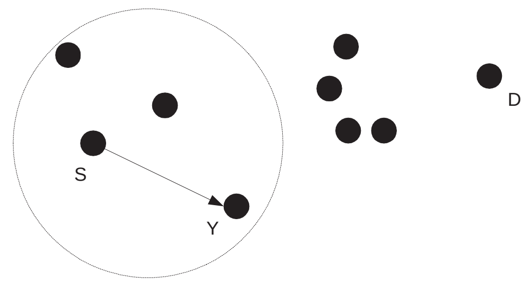


Figure 11: Y is S's closest neighbor in greedy forwarding [32].

GPSR also uses the location of the node to selectively forward the packets based on the distance. The forwarding is carried out on a *greedy* basis by selecting the node closest to the destination (Figure 11). This process continues until the destination is reached. However, in some scenarios, the best path may be through a node which is farther in geometric distance from the destination. In this case, a well known right hand rule is applied to move around the obstacle and resume the greedy forwarding as soon as possible.

Let us note that the location information is shared by beacons from the MAC layer. A node uses a simplistic beaconing algorithm to broadcast beacon packets containing the node ID and its x and y co-ordinates at periodic intervals, helping its neighbors to keep their routing tables updated. With greater mobility, the beaconing interval must be reduced to maintain up-to-date routing tables; however, this results in greater control overhead. To reduce this cost, the sender node's location information is piggybacked with the data packets.

Location Aided Knowledge Extraction Routing for Mobile Ad Hoc Networks (LAKER) [33]: This protocol minimizes the network overhead during the route discovery process by decreasing the zonal area in which route request packets are forwarded. During this process, LAKER extracts knowledge of the nodal density distribution of the network and remember a series of "important" locations on the path to the destination. These locations are named "guiding routes" and with the help of these guiding routes the route discovery process is narrowed down.

LAKER uses the same forwarding strategy as DSR and caches the *forwarding routes* and also creates its own *guiding routes*. While a forwarding route is a series of nodes from the source to the destination, the guiding route contains a series of locations along this route where there may be a cluster of nodes. Even though individual nodes may move around a bit, the basic cluster topology generally remains similar for an extended period of time. Thus, the information found in the first route discovery round is stored and used during the subsequent route discoveries. Since LAKER uses the guiding route caches in route discovery,

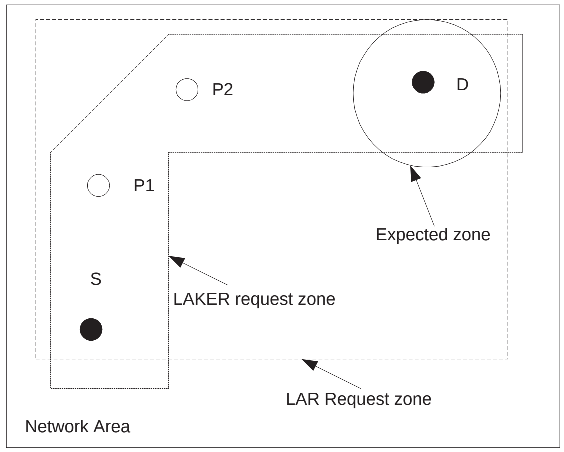


Figure 12: Request zone LAKER vs LAR [33].

the mobility model chosen becomes very important. The restricted random waypoint mobility model [34], which is an extension of [35] is used. Since LAKER is a descendant of DSR and LAR, it uses an on-demand request-reply mechanism for route discovery. The control packet format in LAKER contains both characteristics of DSR and LAR along with its forwarding route and guiding route metrics which aim to decrease the forwarding area of these route request packets as described earlier. Figure 12 shows that the request zone in LAKER is more specific in comparison with LAR; therefore, there is a greater probability of accuracy in determining the exact location of the destination node. Knowledge extraction is carried out by keeping track of the number of neighbors of each node till a certain threshold value is reached. The route reply packet forwards these discovered forwarding and guiding routes to the source.

Movement-Based Algorithm for Ad Hoc Networks (MORA) [36]: In addition to forwarding packets based on the location information, MORA also takes into account the direction of the movement of the neighboring nodes. The metric used for making the forwarding decision is a combination of the number of hops which have an arbitrary weight assigned and a function independent of each node.

While calculating this function F, the primary goal remains to make full use of the directions of the neighboring nodes' movement in selecting the optimal path from source to destination. The function F should depend upon the distance of the node from the line joining the source and destination (sd) and the direction it is moving towards. The function should reach the maxima when the node is moving on sd and should decrease with an increase in the distance from this line. The MORA protocol has two versions:

1. *UMORA*: This is the Unabridged-MORA version since it is very similar to source routing on IP

networks. Here, a short message (called a *probe*) is used to localize the position of the destination. The destination sends a probe along various different routes. Each node receiving this packet keeps updating its own weight function accordingly. After a fixed period of time, the source has all the paths to the destination and corresponding weight functions and selects the most suitable path based on this information.

2. *D-MORA*: This version is the Distributed MORA. It is a scalable algorithm and uses a single path from source to destination. A short probe message is also forwarded from the destination to the source. In every k hops the node receiving the packet polls for information from the neighboring nodes. The packet is then forwarded to the node with the higher link weight. The path information is attached to the packet header and forwarded to the next node.

4.5 Multipath protocols

Multipath routing protocols create multiple routes from source to destination. The main advantage of discovering multiple paths is that the bandwidth between links is used more effectively with greater delivery reliability. It also helps during times of the network congestion. Multiple paths are generated on-demand or using a pro-active approach and is of great significance as routes generally get disconnected quickly due to node mobility.

CacHing And Multipath routing Protocol (CHAMP) [37]: The CHAMP protocol uses data caching and shortest multipath routing. It also reduces packet drops in the presence of frequent route breakages. Every node maintains a small buffer for caching the forwarded packets. This technique is helpful in the case when a node close to the destination encounters a forwarding error and cannot transmit the packet. In such a situation, instead of the source retransmitting again, an upstream node which has a cached copy of the packet may retransmit it, thereby reducing end-to-end packet delay. In order to achieve this, multiple paths to the destination must be available.

In CHAMP each node maintains two caches; a route cache containing forwarding information and a route request cache which contains the recently received and processed route requests. Those entries which have not been used for a specific route lifetime are deleted from the route cache. A node also maintains a send buffer for waiting packets and a data cache for storing the recently forwarded data packets. A route discovery is initiated when there is no available route. The destination replies back with a corresponding route reply packet. There may be multiple routes of equal length established, each with a forwarding count value which

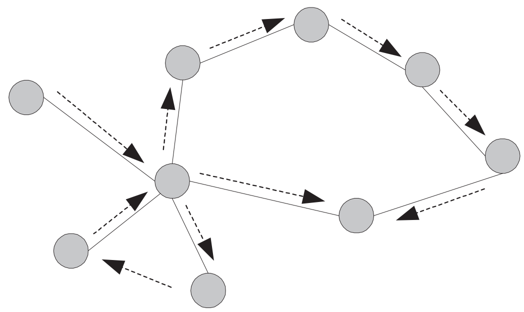


Figure 13: Example of a potential routing loop scenario with multiple path computation [38].

starts with a zero from the source and is increased by one with every retransmission.

Ad hoc On-demand Multipath Distance Vector routing (AOMDV) [38]: The AOMDV protocol uses the basic AODV route construction process, with extensions to create multiple loop-free and link-disjoint paths. AOMDV mainly computes the multiple paths during route discovery process and it consists of two main components: a rule for route updates to find multiple paths at each node, and a distributed protocol to calculate the link-disjoint paths.

In this protocol, each route request and route reply packet arriving at a node is potentially using a different route from the source to the destination. All of these routes cannot be accepted since they can lead to creation of loops (see Figure 13).

The proposed "advertised hop count" metric is used in such a scenario. The advertised hop count for a particular node is the maximum acceptable hop count for any path recorded at that node. A path with a greater hop count value is simply discarded and only those paths with a hop count less than the advertised value is accepted. Values greater than this threshold means the route most probably has a loop.

The following proven property allows to have disjoint routes [38]: *Let a node S flood a packet m in the network. The set of copies of m received at any node I (not equal S) each arriving via a different neighbor of S, defines a set of node-disjoint paths from I to S.* This distributed protocol is used in the intermediate nodes where multiple copies of the same route request packet is not immediately discarded. Each packet is checked to find whether it provides a node-disjoint path to the source.

Split Multipath Routing (SMR) [39]: The SMR protocol establishes and uses multiple routes of maximally disjoint paths from source to destination. Similar to any reactive multipath protocol, multiple routes are discovered on-demand and the route with the shortest delay is selected.

Algorithm 4 AOMDV routing protocol: Route update rules [38]

```
// seqnum(d,i)= Sequence number for destination d at node i
// advertised_hopcount(d,i) = Advertised hop count for d at i
// route_list(d,i) = Route list for d at i

if (seqnum(d,i) is less than seqnum(d,j))
   initialize seqnum(d,i) to seqnum(d,j)
   if (i is not equal d)
      initialize advertised_hopcount(d,i) to infinity
      initialize the route_list(d,i) to NULL
      insert j and advertised_hopcount(d,j) + 1 to route_list(d,i)
   else
      initialize advertised_hopcount(d,i) to 0
   end if
else if ((seqnum(d,i) equals seqnum(d,j)) and
    ((advertised_hopcount(d,i),i) is greater than advertised_hopcount(d,j),j))
   insert j and advertised_hopcount(d,j) + 1 to route_list(d,j)
end if
```

When a node wants to send a packet to a destination for which a route is not known, it floods a RREQ packet into the network. Due to flooding, several duplicate RREQ messages reach the destination along various different paths. Source routing is used since the destination needs to select multiple disjoint paths to send the RREP packet.

Unlike the conventional routing protocols such as AODV and DSR, the intermediate nodes in SMR are not allowed to send back RREPs even if they have the route to the destination. This is because the destination can only make a decision on the validity of maximally disjoint multiple paths from all of its received RREQ packets. If the intermediate nodes reply back, it is almost impossible for the destination to keep track of the routes forwarded to the source. Intermediate nodes also use a different packet forwarding approach. Instead of dropping all duplicate RREQs, each node only forwards those RREQ packets which arrived using a different link from the first RREQ packet and having a hop count lower than the first RREQ packet.

The destination considers the first received RREQ packet as the path with the shortest delay. It immediately sends a RREP back to minimize the route acquisition latency. To find the maximal disjoint path to the already replied route, it waits for additional time to determine all possible route instances. In some cases, there may be more than one maximal disjoint route and if so, the shortest hop distance route is selected.

Neighbor Table Based Multipath Routing (NTBR) [40]: An initial theoretical analysis showing that

Algorithm 5 SMR routing protocol [39]

```
// Source(S) has data to send but has no route to Destination(D)

Transmit a RREQ containing source ID and sequence number
if (receiving node is not D)
   check sequence number for RREQ and incoming link
   if (duplicate packet and different incoming link)
      forward the packet to the neighbor nodes
   else
      drop the packet
   end if
else RREQ at destination
   perform route selection
end if

D receives first RREQ packet
D sends back RREP along the source route in RREQ
D waits for a threshold time duration
At the end of the time interval, D selects the maximally disjoint route from the
route already chosen and the one with the shortest latency
D sends another RREP to S along this new route
S can use any or both routes to send packets
```

non-disjoint multipath routing has a higher route reliability than the conventional disjoint multipath routing led to the development of a neighbor table based multipath protocol which does not require the disjoint routes. In NTBR, every node maintains a neighbor table which records routing information related to its k hop neighbors. k can be set to any value; however, the control overhead also increases with an increase of this value.

The NTBR protocol has a route discovery and route maintenance mechanisms. It also maintains a route cache at each node in the network. The route caches are maintained by the neighbor tables which also serve to make an estimate of the lifetime of the wireless links. This information is used to keep track of the route lifetime. Every node transmits periodic beacon packets to its two-hop neighbors. The neighbor table is established based on the information from the beacon packet by using any of the following approaches:

- *Time driven*: In this approach each node essentially waits for a predefined timeout interval before deciding whether a link is active or not. The node waits for a beacon packet from either its one-hop or two-hop neighbors and adds the information to its neighbor routing table. However, the problem with this approach is that there is always a timeout between the actual topology change and the time in which this information is realized by the node.

- *Data-driven*: This approach alleviates the problem arising from the time driven mechanism. In this scheme, one field of the beacon packet is used to inform whether a node is unreachable or not. The address of the unreachable node is added into the beacon packet and all nodes receiving the packet update their neighbor routing tables accordingly.

The route cache contains all the routing information for a particular node and it is updated by monitoring any packet passing through the network. To extract individual routes, the *route extraction reason* mechanism is used which simply prioritizes the routes extracted from different packets. The routes from route replies are assigned the highest priority while the routes from route request packets or neighbor tables become second followed by the routes from data packets. These priorities are used during the route selection process. The route discovery and route maintenance are similar to the other routing algorithms.

5 Conclusions

In this chapter, we have discussed various routing protocols in ad hoc networks. The common goals of designing a routing algorithm is to reduce control packet overhead, maximize throughput, and minimize the end-to-end delay; however, they differ in ways of finding and/or maintaining the routes between source-destination pairs. We have divided the ad hoc routing protocols into five categories: i) source-initiated (reactive or on-demand), ii) table-driven (proactive), iii) hybrid, iv) location-aware (geographical), and v) multipath. We have then compared the protocols based on common characteristics.

Questions

1. If N is the total number of nodes operating in the network, calculate the control packet overhead complexity and the memory complexity of the following protocols - DSDV, GSR, DREAM and OLSR.
2. Route discovery and route maintenance are two important characteristics of reactive routing protocols. If N is the number of nodes in the network and D is the diameter of the network, calculate the time and communication complexity of both the route discovery and route maintenance operations for the following protocols - AODV, DSR and TORA.
3. It has been shown that the net throughput in DSDV decreases at a greater rate than in AODV. What could be the main reason for this? What other network characteristics can be the deciding factors in

Table 1: Routing protocols comparison

Protocol	Category	Metrics	Route Recovery	Route repository	Loop Free	Communication Overhead	Feature
DSR	Reactive	Shortest path, next available	New route, notify source	Route cache	Yes	High	Completely on demand
AODV	Reactive	Newest route, shortest path	Same as DSR, local repair	Routing table	Yes	High	Only keeps track of next hop in route
TORA	Reactive	Shortest path, next available	Reverse link	Routing table	Yes	High	Control packets localized to area of topology change
ABR	Reactive	Strongest associativity	Local broadcast	Routing table	Yes	Medium	High delays in route repair
SSBR	Reactive	Strongest signal strength	New route, notify source	Routing table	Yes	Medium	Uses a signal stability table
ARA	Reactive	Shortest path	Alternate route, backtrack	Routing table	Yes	Medium	Uses swarm intelligence concepts
ROAM	Reactive	Shortest path	Erase route, start new search	Routing table	Yes	Low	Removes count-to-infinity problem
DSDV	Proactive	Shortest path	Periodic broadcast	Routing table	Yes	High	Distributed algorithm
R-DSDV	Proactive	Shortest path	Randomized updates	Routing table	Yes	Low	Probablistic table updates
OLSR	Proactive	Shortest path	Periodic updates	Routing table	Not always	High	Uses MPRs as routers
CGSR	Proactive	Shortest path	Periodic updates	Routing table	Yes	Low	Clusterhead is critical node
WRP	Proactive	Shortest path	Periodic updates	Routing table	Yes	Low	Uses HELLO messages
STAR	Proactive	Shortest path	Specific updates	Routing table	Yes	Low	Updates at specific events
ZRP	Hybrid	Shortest path	Start repair at failure point	Interzone, intrazone tables	Yes	Medium	Routing range defined in hops
FSR	Hybrid	Scope range	Notify source	Routing tables	Yes	Low	Updates are localized
LANMAR	Hybrid	Shortest path	Notify source	Routing table at landmark	Yes	Medium	Using landmarks increases scalability
RDMAR	Hybrid	Shortest path	New route, notify source	Routing table	Yes	High	Localized query flooding
SLURP	Hybrid	MFR for interzone, DSR for interzone	Notify source	Route cache at location	Yes	High	Eliminates global route discovery
A4LP	Hybrid	Power consumed	Notify source	Routing table	Yes	Medium	Uses asymmetric links
LAR		Hop count	Notify source	Route cache	No	Medium	RERR message on link break
DREAM	Geographical	Hop count	Any available method	Routing table	No	Low	Location table at each node
GPSR	Geographical	Shortest path	N/A	Route cache	Yes	High	Greedy and perimeter forwarding
LAKER	Geographical	Hop count	Notify source	Route cache	No	Low	Knowledge guided route discovery
MORA	Geographical	Weighted hop count	New route, notify source	Routing table	No	High	RREQ based on metric 'm'
CHAMP	Multipath	Shortest path	Notify source	Route cache	Yes	High	Performs load balancing
AOMDV	Multipath	Advertised hop count	Local repair	Routing table	Yes	High	Multipath extensions to AODV
SMR	Multipath	Least delay	Notify source	Routing table	Yes	High	2 disjoint routes chosen
NTBR	Multipath	Link active	Notify source	Route cache	Yes	High	Non-disjoint routes selected

determining the effectiveness of deploying one of these protocols in a real ad hoc network?

4. DSDV and AODV are both deployed on an experimental network under the same network characteristics for a comparative study. Which protocol will have higher bandwidth consumption and why? Note that DSDV sends periodic route broadcasts while AODV carries out route maintenance using periodic "HELLO" packets. Considering this property, which protocol would accumulate a greater control overhead?

5. List the major points of distinction between the LAR and the DREAM routing protocols. LAR routes packets by using the GPS system to determine the location of the nodes in the network. However, does it take into account the presence of obstructions in the network area? Will such obstructions degrade the protocol's performance or is there any technique to route "around" these obstructions?

6. Describe the primary disadvantages of multipath approaches to routing in comparison with unipath protocols. Multipath reactive protocols do not allow intermediate nodes to reply to RREQ packets. Why? Is it beneficial for the network? Explain your reasoning.

7. Multipath routing brings in the traffic allocation problem. If a source has a set of paths to the destination, how should it allocate traffic to each of the different routes? Traffic allocation is the job of the transport layer, but there seems to be a need of some cross-layer interactions in determining the appropriate route and send the correct amount of traffic on most suitable routes. Stronger links should be allocated more traffic than other weaker links. Discuss probable solutions and issues in determining the appropriate traffic quantities in the presence of multiple paths.

8. Routing is carried out under a varied set of constraints and requirements. Candidate selection for a node has been shown to be an NP-Complete problem. It is almost impossible to optimize all parameters (for example hop count, control packet overhead and link quality among others) when a route is selected. Hence, what should be the major design goal a researcher who proposes a new routing protocol for ad hoc networks?

References

[1] C. Perkins (Ed.), Ad hoc Networking, Addison Wesley, 2001.

[2] M. Weiser, Some computer science issues in ubiquitous computing, ACM SIGMOBILE Mobile Computing and Communications Review 3 (3) (1999) 12.

[3] B. Bollobas, Random Graphs, Academic Press, 1985.

[4] D. Johnson, D. Maltz, Dynamic source routing in ad hoc wireless networks, in: T. Imielinski, H. Korth (Eds.), Mobile Computing, Kluwer Academic Publishers, 1996, Ch. 5, pp. 153–181.

[5] C. Perkins, E. Royer, Ad hoc On-Demand Distance Vector Routing, in: Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications, 1999, pp. 99–100.

[6] C. Perkins, P. Bhagwat, Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers, in: Proceedings of the ACM SIGCOMM, 1994, pp. 234–244.

[7] V. Park, M. Corson, A highly adaptive distributed routing algorithm for mobile wireless networks, in: Proceedings of the IEEE INFOCOM, 1997, pp. 1405–1413.

[8] M. S. Corson, A. Ephremides, A distributed routing algorithm for mobile wireless networks, ACM/Baltzer Wireless Networks Journal 1 (1) (1995) 61–81.

[9] Temporally ordered routing algorithm (TORA), http://wiki.uni.lu/secan-lab/Temporally-Ordered+Routing+Algorithm.html.

[10] C.-K. Toh, Associativity-based routing for ad hoc mobile networks, Wireless Personal Communications Journal, Special Issue on Mobile Networking and Computing Systems 4 (2) (1997) 103–139.

[11] R. Dube, C. Rais, K. Wang, S. Tripathi, Signal stability-based adaptive routing (SSA) for ad hoc mobile networks, IEEE Personal Communications Magazine (1997) 36–45.

[12] M. Gunes, U. Sorges, I. Bouazizi, ARA–the ant-colony based routing algorithm for manets, in: Proceedings of the IEEE ICPP Workshop on Ad Hoc Networks (IWAHN), 2002, pp. 79–85.

[13] J. Raju, J. Garcia-Luna-Aceves, A new approach to on-demand loop-free multipath routing, in: Proceedings of the IEEE International Conference on Computer Communications and Networks (IC3N), 1999, pp. 522–527.

[14] J. Garcia-Luna-Aceves, Loop-free routing using diffusing computations, IEEE/ACM Transactions on Networking 1 (1) (1993) 130–141.

[15] A. Boukerche, S. K. Das, A. Fabbri, Analysis of a randomized congestion control scheme with DSDV routing in ad hoc wireless networks, Journal of Parallel and Distributed Computing 61 (7) (2001) 967–995.

[16] T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, A. Qayyum, L. Viennot, Optimized link state routing protocol for ad hoc networks, in: Proceedings of the IEEE INMIC, 2001, pp. 62–68.

[17] C. Chiang, H. Wu, W. Liu, M. Gerla, Routing in clustered multihop, mobile wireless networks, in: Proceedings of IEEE SICON, 1997, pp. 197–211.

[18] S. Murthy, J. Garcia-Luna-Aceves, An efficient routing protocol for wireless networks, MONET 1 (2) (1996) 183–197.

[19] J. Garcia-Luna-Aceves, M. Spohn, Source-tree routing in wireless networks, in: Proceedings of the IEEE ICNP, 1999, pp. 273–282.

[20] S. Roy, J. Garcia-Luna-Aceves, Using minimal source trees for on-demand routing in ad hoc networks, in: Proceedings of the IEEE INFOCOM, 2001, pp. 1172–1181.

[21] P. Samar, M. Pearlman, S. Haas, Independent zone routing: an adaptive hybrid routing framework for ad hoc wireless networks, IEEE/ACM Transactions on Networking 12 (4) (2004) 595–608.

[22] G. Pei, M. Gerla, T.-W. Chen, Fisheye state routing in mobile ad hoc networks, in: Proceedings of ICDCS Workshop on Wireless Networks and Mobile Computing, 2000, pp. D71–D78.

[23] L. Kleinrock, K. Stevens, Fisheye: A lenslike computer display transformation, Tech. rep., Dept. of Computer Science, UCLA (1971).

[24] G. Pei, M. Gerla, X. Hong, LANMAR: landmark routing for large scale wireless ad hoc networks with group mobility, in: Proceedings of the ACM MobiHoc, 2000, pp. 11–18.

[25] G. N. Aggelou, R. Tafazolli, RDMAR: a bandwidth-efficient routing protocol for mobile ad hoc networks, in: Proceedings of the IEEE WoWMoM, 1999, pp. 26–33.

[26] S.-C. Woo, S. Singh, Scalable routing protocol for ad hoc networks, Wireless Networks 7 (5) (2001) 513–529.

[27] G. Wang, Y. Ji, D. C. Marinescu, D. Turgut, A routing protocol for power constrained networks with asymmetric links, in: Proceedings of the ACM Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN), 2004, pp. 69–76.

[28] G. Wang, D. Turgut, L. Bölöni, Y. Ji, D. Marinescu, Improving routing performance through m-limited forwarding in power-constrained wireless networks, To Appear in Journal of Parallel and Distributed Computing (JPDC) .

[29] Y. Ko, N. Vaidya, Location-Aided Routing (LAR) in mobile ad hoc networks, in: Proceedings of the ACM MOBICOM, 1998, pp. 66–75.

[30] Y. Ko, N. Vaidya, Location-Aided Routing (LAR) in mobile ad hoc networks, Wireless Networks 6 (4) (2000) 307–321.

[31] S. Basagni, I. Chlamtac, V. Syrotiuk, B. Woodward, A distance routing effect algorithm for mobility (DREAM), in: Proceedings of the ACM MOBICOM, 1998, pp. 76–84.

[32] B. Karp, H. Kung, GPSR: greedy perimeter stateless routing for wireless networks, in: Proceedings of the ACM MOBICOM, 2000, pp. 243–254.

[33] J. Li, P. Mohapatra, LAKER: location aided knowledge extraction routing for mobile ad hoc networks, in: Proceedings of the IEEE WCNC, 2003, pp. 1180–1184.

[34] L. Blazevic, S. Giordano, J.-Y. L. Boudec, Self organized routing in wide area mobile ad-hoc network, in: Proceedings of the IEEE GLOBECOM, 2001, pp. 2814–2818.

[35] J. Broch, D. Maltz, D. Johnson, Y.-C. Hu, J. Jetcheva, A performance comparison of multi-hop wireless ad hoc network routing protocols, in: Proceedings of the ACM MOBICOM, 1998, pp. 85–97.

[36] G. Boato, F. Granelli, MORA: a movement-based routing algorithm for vehicle ad hoc networks, in: Proceedings of the International Conference on Distributed Multimedia Systems (DMS), 2004, pp. 171–174.

[37] A. Valera, W. K. G. Seah, S. V. Rao, Cooperative packet caching and shortest multipath routing in mobile ad hoc networks, in: Proceedings of the IEEE INFOCOM, Vol. 1, 2003, pp. 260–269.

[38] M. K. Marina, S. Das, On-demand multi path distance vector routing in ad hoc networks, in: Proceeding of the IEEE ICNP, 2001, pp. 14–23.

[39] S. Lee, M. Gerla, Split multipath routing with maximally disjoint paths in ad hoc networks, in: Proceedings of the IEEE ICC, Vol. 10, 2001, pp. 3201–3205.

[40] Z. Yao, J. Jiang, P. Fan, Z. Cao, V. Li, A neighbor-table-based multipath routing in ad hoc networks, in: Proceedings of the IEEE VTC 2003-Spring, Vol. 3, 2003, pp. 1739–1743.