Routing protocols in ad hoc networks: a survey

Azzedine Boukerche, Begumhan Turgut, Nevin Aydin, Mohammad Z. Ahmad, Ladislau Bölöni, and Damla Turgut[1]

Abstract

Ad hoc wireless networks perform the difficult task of multi-hop communication in an environment without a dedicated infrastructure, with mobile nodes and changing network topology. Different deployments exhibit various constraints, such as energy limitations, opportunities, such as the knowledge of the physical location of the nodes in certain scenarios, and requirements, such as real-time or multi-cast communication. In the last 15 years, the wireless networking community designed hundreds of new routing protocols targeting the various scenarios of this design space. The objective of this paper is to create a taxonomy of the ad hoc routing protocols, and to survey and compare representative examples for each class of protocols. We strive to uncover the requirements considered by the different protocols, the resource limitations under which they operate, and the design decisions made by the authors.

Key words: ad hoc networks, sensor networks, routing protocols

1. Introduction

Wireless local area networks based on the 802.11a,b,g and n standards became one of the most ubiquitous ways of networking with mobile nodes. Most of these networks, however, are deployed in the configuration which can be called "wired everywhere, except the first hop". If the goal of the user of the mobile computer is to connect to a website located halfway around the world, the best strategy is to escape as quickly as possible from the challenges of wireless domain and enter the reliability of fiber optic networks and time-tested networking protocols. In such networks, all the nodes connect to an *access point* which usually has a wired connection to the Internet. From the point of view of the network and higher layers, this first hop can be approximated as an Ethernet-type shared medium. In this scenario the nodes connected to the same wireless LAN communicate with each other only indirectly.

There are, however, many important applications where this model is not applicable. First, even if the goal is Internet access, the access point might not be able to cover all the relevant mobile nodes due to limitations in transmission range, cost or access rights considerations. Another case is when Internet access is not desired (or is secondary importance), the main application being to communicate locally among a group of (potentially mobile) nodes.

These scenarios can be serviced only if we allow some (possibly all) routing hops to be performed in the wireless domain. Such networks can be set up in any location in an *ad hoc* manner, without the need of an existing wired infrastructure.

These networks are known as *ad hoc wireless networks* [92], other proposed names being *infrastructureless wireless networks*, *instant infrastructure* [8] and *mobile-mesh networking* [140].

One of the major technological challenges of such networks is that they require new types of routing protocols. As opposed to the wired infrastructure, there are no dedicated router nodes: the task of routing needs to be performed by the user nodes, which can be mobile, unreliable and have limited energy and other resources.

The goal of this paper is to review the collection of technologies which have been proposed for routing in ad hoc networks. There are literally hundreds of different ad hoc routing protocols proposed. We strive not for a simple enumeration of this extensive literature, but we try to uncover the design decisions behind the various protocols, their interrelationship, and the specific requirements taken into consideration by the designers.

This way, we hope to provide the student and researcher with a more clear description of the state of the art. We hope that this systematic approach will help the researcher understand the open challenges of the field, as well as those which have been satisfactorily solved. This can help a researcher position its work in the context of the state of the art, assess its originality and avoid duplication of existing

work. The survey can also help the practitioners choose the most adequate technology for a specific deployment.

Early ad hoc routing protocols have been classified into *on-demand* and *table-driven* protocols. Between these two, several hybrid approaches have been developed. The increasing size of the ad hoc networks considered made necessary the use of techniques such as geographical routing and hierarchical routing. Finally, in many deployments of ad hoc networks, the problem of energy conservation takes precedence from all the other performance metrics, thus power aware routing protocols will be treated as a separate class.

2. Applications of ad hoc networks

Ad hoc networks, defined in the broad sense by of the term as wireless networking in the absence of a wired infrastructure, have a wide range of potential applications. Some of these applications have been already identified in early ad hoc literature[92]. Other applications, however, were enabled only recently by the shifting landscape of computing and communication. In the last decade, for instance, the default method for first hop internet access shifted from the Ethernet line to the WiFi connection. In the last three years, smartphones and tablets emerged as a multi-purpose computing platform, relying exclusively on wireless connectivity and replacing personal digital assistants and (in some cases) low-end mobile computers. These technological advances have enabled applications such as urban sensing which could not have been predicted ten years ago. On the other hand, some wireless technologies progressed at a slower pace than predicted: vehicle to vehicle technologies are still in the prototype phase, home automation systems are rare, and sensing devices are still too expensive to be considered disposable and too large to be deployed by random spreading.

In the following we briefly survey some application areas of interest for ad hoc networks. There are some applications where ad hoc networks are the only possible solution, for instance, networking in areas where no infrastructure is available. Early work in ad hoc networks frequently assumed such radical scenarios. Beyond these applications, however, there is a much larger field of potential applications where ad hoc networks compete with other possible technical solutions. Finally, there are application areas where ad hoc networks must be part of a combination of technologies.

Network extension: In this application area, the networking infrastructure exists, but it has insufficient coverage. The goal of the participants of the network is *internet access*, that is, their main communication partners are outside the ad hoc network. The goal of the ad hoc network is to extend the internet connectivity beyond the reach of the access points. Most routes of the ad hoc network will connect the access points to the nodes.

Local interconnection networks: In this application area, no infrastructure is available (or the nodes choose not to use it). For instance, when networking in remote areas (such as in a scenario involving a camp of archaeologists in the Central American forest), the infrastructure might not have been there to begin with. In other applications, such as disaster response, the previously existing infrastructure has collapsed due to a natural disaster. In these applications, the communication partners of most nodes are *within the network*. Example applications include point-to-point messaging and audio and video conferencing.

Ubiquitous computing: This area covers networking between devices embedded in the environment. Communication patterns in ubiquitous computing are strongly influenced by the physical location and proximity - devices which are close to each other are more likely to communicate then remote devices. In contrast, on the wired internet, physical location is almost irrelevant. Ad hoc networks are a particularly good match for proximity based communication. Note, however, that in areas where a pervasive infrastructure is available, ad hoc networks compete with solutions which rely on the convenience of the default infrastructure, even when technologically suboptimal. A recent example involves solutions where a TV set-top box is controlled from a smartphone, through an internet connection traversing dozens of routers, even when the two devices are several feet from each other.

Urban sensing: This application area exploits the sensing and computation capabilities of smartphones, together with the wide range of their deployment in urban areas. Urban sensing is characterized by distributed sensing or data collection, and, in many cases, by distributed data customers. Smartphones can use both infrastructure based access and as well as ad hoc connections. Ad hoc approaches have the

advantage of lower energy consumption, lower overall bandwidth consumption and improved privacy - but they inevitably involve more complex interaction patterns.

Vehicular networking: This area covers applications where one of the communication partners is a vehicle. This definition covers a very wide range of technologies.

One type of vehicular networking involves the short range communication between the vehicle and personal devices carried by the passengers. The most widely deployed systems rely on Bluetooth, but other technologies, such as WiFi have also been suggested and implemented. Note that many devices, while connected to the local network of the vehicle, can also maintain long range wireless connections, for instance through 3G cellular telephony, and, increasingly, WiMAX and LTE.

Another area of vehicular networking is vehicle to vehicle (V2V) communication. Due to the high relative speed of the participants, the short duration of the encounters and the peer-to-peer nature of the communication, V2V technologies are a good match for ad hoc networking. V2V has applications in convoy driving, accident pre-emption, and lane changes with pre-negotiated lane clearance. One significant obstacle of the deployment of V2V technologies relate to achieving a critical mass of deployment - a single V2V-capable surrounded with vehicles which cannot understand it cannot achieve any of the benefits of the technology.

A final area of vehicular networking is vehicle-to-infrastructure (V2I) communication. Calling a V2I system an "ad hoc network" appears to be self-contradictory - as we defined the lack of infrastructure the defining characteristic of ad hoc networks. Yet, vehicles have a high relative speed with respect to any specific access point in the infrastructure - the communication needs to be either very brief, or the system needs to handle very fast connection hand-offs, or it needs to be able to communicate through intermittent connections. Thus, the infrastructure in V2I communications has more in common with the typical mobile interaction partner in ad hoc networks, than the highly reliable, quasi-permanent connections characterizing infrastructure in traditional networking. The most popular applications of V2I are currently toll collection systems. However, these systems will also offer significant support for the emerging intelligent transportation systems.

Personal area networks: This application area refers to networking among the portable devices carried by a single user. As long as these devices move with the user, this system can be considered as a local area network with the individual components being in a fixed relative position. The most popular current PAN technology is based on the Bluetooth standards. Quite often, one or more of these devices has its own internet connectivity through long range communication. Very similarly with the vehicular networks, aspects of ad hoc networking come into play when the personal networks of different users will need to interact, or when devices of one users' PAN needs to establish a short range communication with infrastructure elements (such as when performing payment processing through near-field communication).

3. Ad hoc routing protocols and comparisons

Mirroring the diversity of applications areas, researchers have proposed a wide range of routing protocols for ad hoc networks. The basic goals of these protocols are the same: maximize throughput while minimizing packet loss, control overhead and energy usage. However, the relative priorities of these criteria differ among application areas. In addition, in some applications, ad hoc networking is really the only feasible solution, while in other applications, ad hoc networking competes with other technologies. Thus, the performance expectations of the ad hoc networks differ from application to application and the architecture of the ad hoc network, thus each application area and ad hoc network type must be evaluated against a different set of metrics (although some metrics can be applied across several protocol categories).

In the reminder of this paper, we organize the discussed routing protocols into nine categories based on their underlying architectural framework as follows (also shown in Figure 1).

- Source-initiated (Reactive or on-demand) (Section 3.1)
- Table-driven (Pro-active) (Section 3.2)
- Hybrid (Section 3.3)
- Location-aware (Geographical) (Section 3.4)
- Multipath (Section 3.5)

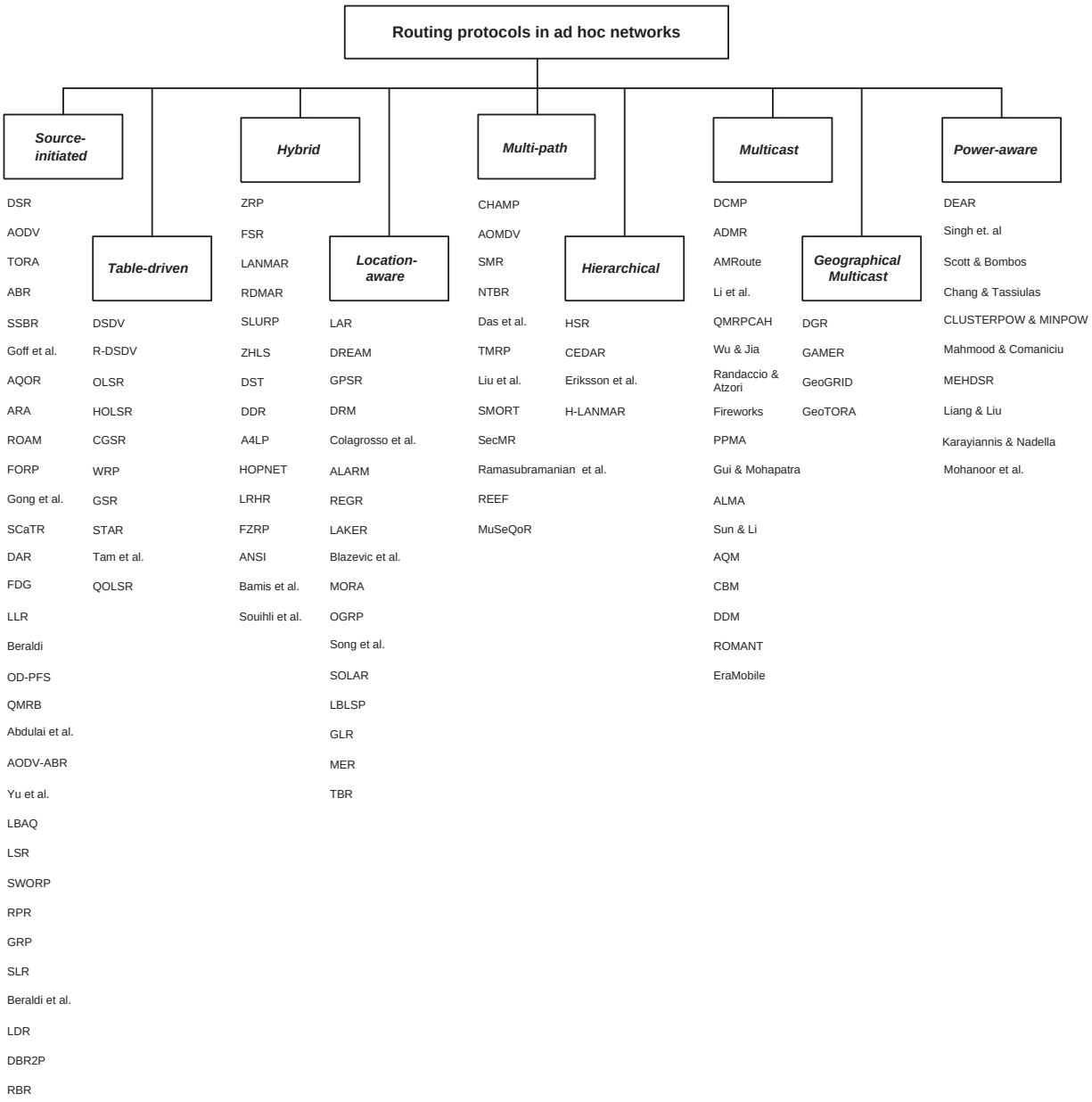


Figure 1: Categories of ad hoc routing protocols

- Hierarchical (Section 3.6)
- Multicast (Section 3.7)
- Geographical Multicast (Section 3.8)
- Power-aware (Section 3.9)

3.1. 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. The route is created through a route discovery procedure which involves flooding the network with route request packets are flooded to starting with the immediate neighbors of the source. Once a route is formed or multiple routes are obtained to the destination, the route discovery process comes to an end. A route maintenance procedure maintains the

continuity of the route for the timespan it is needed by the source.

Dynamic Source Routing (DSR) [57]: Johnson et al. propose one of the most widely known routing algorithms, called Dynamic Source Routing which is an "on-demand" algorithm and it has *route discovery* and *route maintenance* phases.

Route discovery contains both *route request* and *route reply* messages. In the 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 the destination or 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, 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 the source to the 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 the route maintenance phase, *route error* and *acknowledgements* packets are used. 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. Upon receiving the route error message, nodes remove from their route caches the route entry using the broken link.

Ad hoc On-Demand Distance Vector (AODV) [94]: The AODV routing protocol was developed by Perkins and Royer as an improvement to the Destination-Sequenced Distance-Vector (DSDV) routing algorithm [93]. AODV aims to reduce the number of broadcast messages forwarded 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 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. The route request packet contains the IP address of the source node, current sequence number, the IP address of the destination node and the last known sequence number. Figure 2 illustrates the forward and reverse path formation in the AODV protocol. An intermediate node can reply to the route request packet only if it has 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 arrived. 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 `RREP` 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 destination or the intermediate node moves away and it is performed by sending 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.

Temporally Ordered Routing Algorithm (TORA) [89, 139]: Park and Corson proposed TORA, an adaptive and scalable routing algorithm based on the concept of link reversal. It finds multiple routes from a source to a destination in a highly dynamic mobile networking environment. In TORA control messages are localized to a small set of nodes nearby the topological change. Nodes maintain routing information about their immediate one-hop neighbors. The three basic functions of the protocol are route creation, route maintenance, and route erasure.

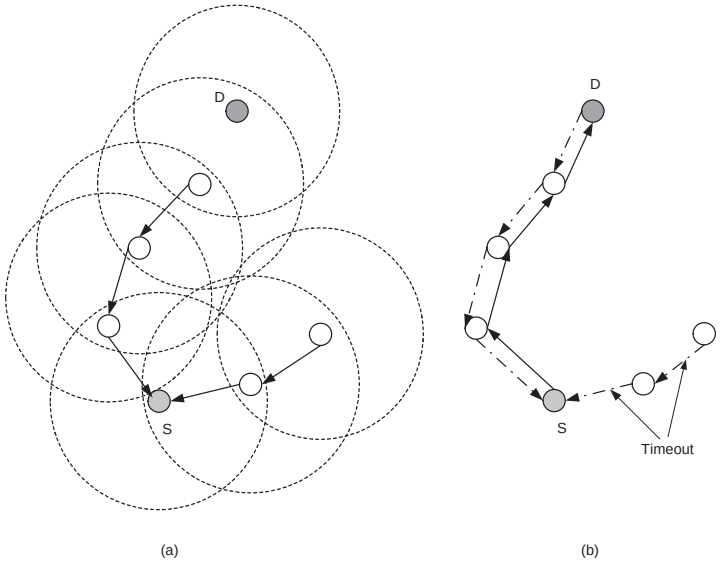


Figure 2: (a) Reverse path formation: the reverse path is set up from all the nodes back to the source through `RREQ` message traveling in the network. (b) Forward path formation: the forward path is constructed as the `RREP` message from the destination node D to source node S. The source node start sending data when it received the first `RREP` message to the sender [94].

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 the unique node ID, the 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 in Lightweight Mobile Routing (LMR) [34]. Route maintenance is necessary when any of the links in DAG is broken. Figure 3 describes the control flow of route maintenance in TORA.

The main strength of the TORA protocol is its approach to handling the link failures. TORA's reaction to link failures is *optimistic:* 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 use a two-passes search for the same task, while both DSR and AODV use a 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) [116]: Toh proposes the ABR algorithm which considers route stability as the most important factor in selecting a route. Routes are discovered by broadcasting a *broadcast query* request packet. Using these packets, the destination becomes aware of all possible routes between itself and the source.

The ABR algorithm maintains a "degree of associativity" by using a mechanism called *associativity ticks*. Each node maintains a tick value for each neighbors, which is increase by one every time a periodic link layer `HELLO` message is received from the neighbor. Once the tick value reaches a specified threshold value, it means that the route is *stable*. If the neighbor goes out 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.

Signal Stability-Based Adaptive Routing (SSBR) [37]: Dube et al. propose the SSBR protocol in which the main routing criteria are the signal and location stability. As in other on-demand routing

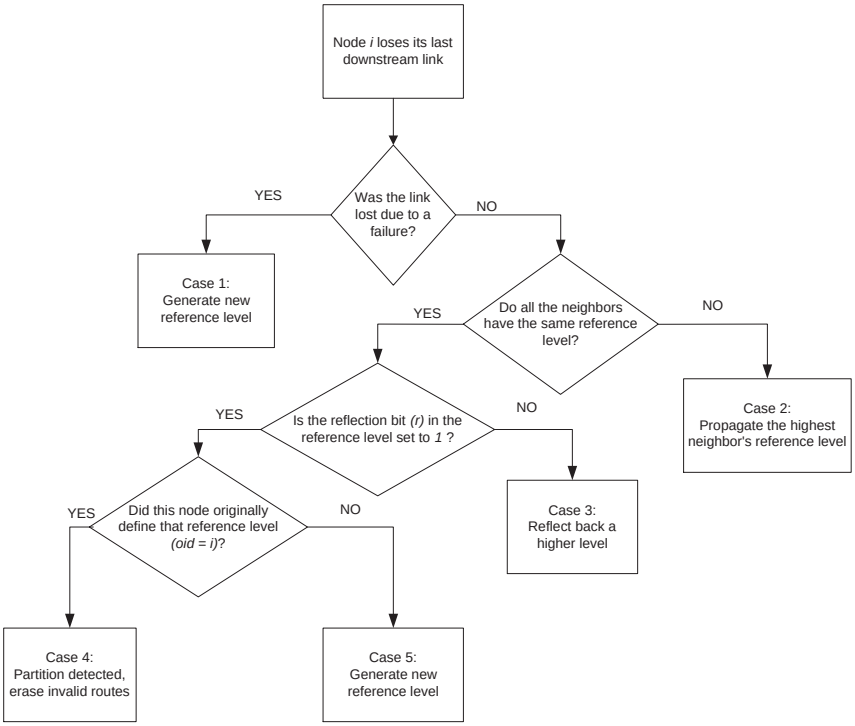


Figure 3: Flow diagram of route maintenance in TORA [89]. The only routes with a height greater than NULL value are maintained. The figures shows all five possible paths in the form of a decision tree which is based on the state of the node and the preceding event.

protocols, the route request is broadcast throughout the network, the destination replies with the route reply message and then the sender sends data through the selected route. Additionally, the signal strength (link quality) between neighboring nodes plays a major role in the route selection process in this protocol.

SSBR consists of two sub-protocols: the *Dynamic Routing Protocol (DRP)* and the *Static Routing Protocol (SRP)*. DRP interacts with the network interface device driver 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 (or it might simply choose the first path it receives). If no strong links are found, the protocol could fall back on other available weaker links.

The paper also discusses two further enhancements to the route selection process. The first selection concerns the selection of alternative routes from a set which In the first case, the link strength 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 a gracious route reply for a route it already has a prior information about.

Preemptive routing in ad hoc networks [47]: In conventional protocols, a path is considered broken only after several retransmissions have timed out. The algorithm introduced by Goff et al. attempts to

initiate the discovery process of an alternate route just before the probable route failure. The algorithm generates a preemptive warning when the signal power of the packet received drops below a predefined *preemptive threshold*. The correct setting of the preemptive threshold is the main challenge of the algorithm. If the value is too high, unnecessary warnings may be generated which can lead to greater overhead, unnecessary route discoveries, and switches to possibly lower quality paths. On the other hand, if the value is too low, the path breaks much earlier than the alternate route is selected. This leaves a short time period for building an alternate path. As temporary channel fading can often create a weak reception without leading to route failure, the algorithm uses successive "query" packets to decide whether the generated warnings are valid.

Ad hoc QoS on-demand routing (AQOR) [132]: Xue and Ganz propose AQOR, an on-demand routing protocol enabling QoS support in terms of bandwidth and end-to-end delay. The AQOR mechanism estimates the bandwidth and end-to-end delay requirements and use these metrics to make admission and resource reservation decisions.

The AQOR integrates on-demand route discovery, signaling functions for resource reservations, and hop-by-hop routing to provide QoS support in ad hoc networks. Since most QoS violations are detected at the destination node, the routing overhead generated can be reduced by initiating the route recovery process at the destination node. Route maintenance is accomplished by sending periodic `HELLO` messages. Routes are discovered on-demand by a limited flooding mechanism. The requested bandwidth and end-to-end delay values are specified within the route request packet. The bandwidth requirements are calculated based on the available link capacity and the bandwidth used by the flow. If this request is accepted, the node updates its routing table with an *explored* status and broadcasts it to all its neighbors. However, if no reply is received in a specified time, the route entry is removed from the node's table and late-arriving replies are simply ignored. Route caching is not used since the route request packets are forwarded from the node to the destination to determine the bandwidth and end-to-end delay requirements. To further reduce control overhead, the packets have a time-to-live (TTL) parameter which stops the packets from traveling unnecessarily throughout the network.

ARA-The Ant-colony based Routing Algorithms [50]: Gunes et al. present a novel technique for ad hoc routing by using concepts of *swarm intelligence* and the *ant colony* meta-heuristic. This class of algorithms aims to solve the complex optimization and collaboration problems without direct communication among the participants. Indirect communication is achieved by *stigmergy*, the process of leaving traces in the environment, similar to the behavior of ants leaving pheromone signals.

The route discovery 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. 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 can be used. Figure 5 and Figure 6 show the forward and backward ants' route discovery phases.

The ARA algorithm doesn't need special route maintenance packets since 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 forwarded along the path, but decreases in time when no packets are transmitted.

Routing On-demand Acyclic Multipath (ROAM) [98]: ROAM algorithm by Raju and Garcia-Luna-Aceves coordinates among nodes in directed acyclic subgraphs. It is an extension of the DUAL [41] routing algorithm. The ROAM algorithm guarantees that route search query will fail to return a destination path only is all the routers agree 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 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. 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.

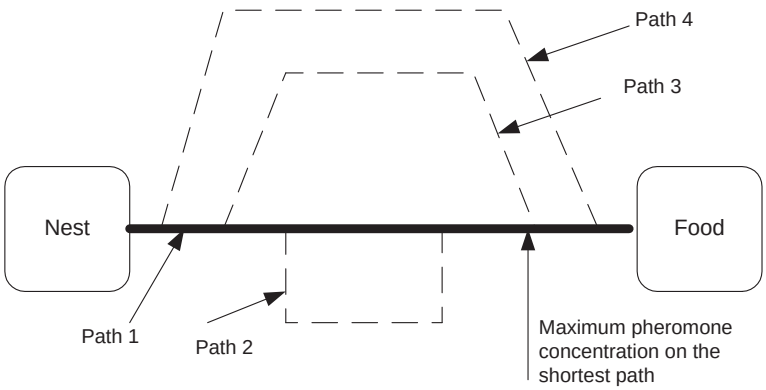


Figure 4: Pheromone concentration along the shortest path used by ants to discover food from their nest [50].

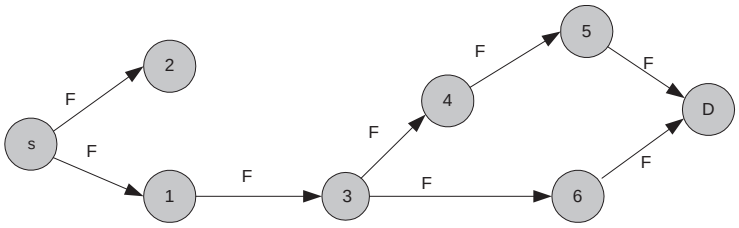


Figure 5: Forward ant route discovery phase. A forward ant (F) is send from the sender (S) toward the destination node (D). The forward ant is relayed by other nodes, which initialize their routing table and the pheromone values [50].

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 to select a neighbor as its successor only if it is a *feasible successor*. This provides a shortest loop-free path to the destination. A when it requires a path to a destination, the source router starts a diffusion search, with the packet propagated through routers which have no entry of the node. The first router with an available route to the destination responds to the source with the distance to the node. At the end of the search, either the source has a finite distance to the destination or the destination is unreachable.

The Flow Oriented Routing Protocol (FORP) [113]: The FORP protocol proposed by Su and Gerla aims to transmit real-time data streams in ad hoc networks, which require in-order delivery of packets with tight delivery bounds. If alternate routes are not available to immediately redirect the data packets in case of route failures, real-time packets may be dropped. FORP introduces the "multi-hop handoff" mechanism in which the nodes use their mobility information to determine future route changes resulting in rebuilding of an alternate routes much sooner.

Similarly to other on-demand schemes, FORP maintains routing information only for active

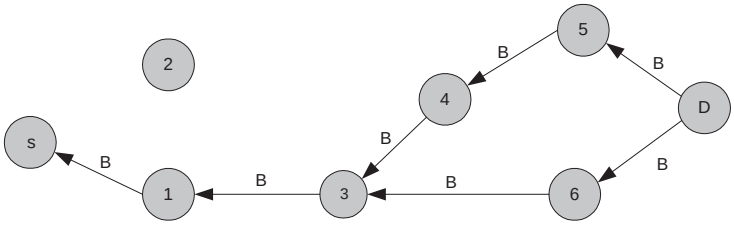


Figure 6: Backward ant route discovery phase. The backward ant (B) has the same task as the forward ant. It is send by the destination node toward the source node [50].

source/destination pairs. The protocol predicts the link expiration time (LET) for each hop on the route to calculate the route expiration time (RET). During the route discovery phase, the source broadcasts a `Flow-REQ` message containing a sequence number, source ID, and destination ID. Each node appends its own ID and the LET of the last link in which the message was received before forwarding to the next hop. When the `Flow-REQ` arrives at the destination, it contains the list of all the routes travelled and the LETs for each hop. Using this information, the destination calculates the RET by selecting the minimum LET value for the route. The nodes are assumed to have a common time reference from GPS for instance. Once the route is selected, a `Flow-SETUP` message travels to the source along the chosen path.

While the connection is in progress, the intermediate nodes continue adding the LETs to the forwarded packets to enable the destination to keep track of the RET prediction. If the destination determines that a "critical time" is reached, i.e. the route is close to expire, a `Flow-HANDOFF` message is generated and propagated throughout the network. These `Flow-HANDOFF` messages reach the source and based on their LETs and RET, the source determines the new route. The "critical time" is calculated as T_c = RET - T_d where RET is the route expiration time and T_d is the delay experienced by the last packet arrived on the same route.

On-demand routing and channel assignment in multi-channel mobile ad hoc networks [48]: Gong et al. concentrates mainly on designing an efficient channel assignment algorithm at the MAC layer to be used with most on-demand routing strategies at the network level. The authors state that there are *intra-flow* and *inter-flow* interferences due to adjacent nodes on the same or different channels respectively. To mitigate the interference problems, the authors implement two enhanced versions of the AODV routing protocol: Enhanced 2-hop CA-AODV and Enhanced k-hop CA-AODV. In these algorithms, the `RREQ` and `RREP` messages are similar to AODV with the difference of the neighbor table at each node consisting of both the route and the indices of the channels already chosen. If the node is not assigned to a channel, a channel is randomly chosen and assigned from the available list (see Figure7). `RREP` messages also carry channel information which is updated by each node receiving the `RREP`. At the end of the `RREQ-RREP` cycle, every node in the route is assigned a channel which is different from any of its k-hop (for the k-hop extension of the algorithm) neighbors on the same route.

Space-Content adaptive Time Routing (SCaTR) [17]: Boice et al. present a routing framework which takes into consideration the possibility of intermittent connectivity in a mobile ad hoc network. SCaTR uses past connectivity information by defining *proxy* nodes to route traffic towards the destination when no direct route is available. It is built upon the existing AODV protocol in such a way that when the network is fully connected, it works identical to AODV. When a network partition occurs, proxy nodes are created based on the distance of a node to the destination. A node closer to the destination advertizes itself as the *proxy destination* and buffers messages on the route until the final destination is discovered or another node is selected as a better proxy. This design feature enables the protocol to behave no worse than the standard AODV in most cases (when the network is entirely connected). During the route discovery phase if a route to the destination is not established, a proxy request (`PREQ`) is forwarded to find the nearest proxy destination. `PREQ` could also request for multiple destinations and a proxy reply (`PREP`) is sent back by the responding node. Each proxy node has a finite buffer to store different messages from varying source-destination pairs. The `PREP` message contains updated information of the proxy's contact value for the destination, its remaining buffer space, and number of messages stored for the particular source-destination pair from which it has received the `PREQ`. Each node also has a separate buffer for its own messages. The source node collects all the `PREP`s, compares the contact values obtained and updates its table with the highest value along with the route where the data packets are forwarded.

Distributed Ant Routing (DAR) [104]: Rosati et al. propose a distributed routing algorithm based on ant behavior in colonies. Ant colony optimization algorithms have been widely used in MANETs and the authors aim to design an algorithm incorporating the salient features of many existing approaches. The main design goal of DAR is to minimize the computation complexity. Each node contains routing tables which are stochastic with the next hop being selected based on weighted probabilites. These probabilities are calculated based on pheromone trails left by ants. Forward ants are used to find new paths. If multiple paths are available at a node, the next hop could be selected either randomly or the most optimal one. DAR mostly uses hop-by-hop optimal forwarding with the forward ant routed

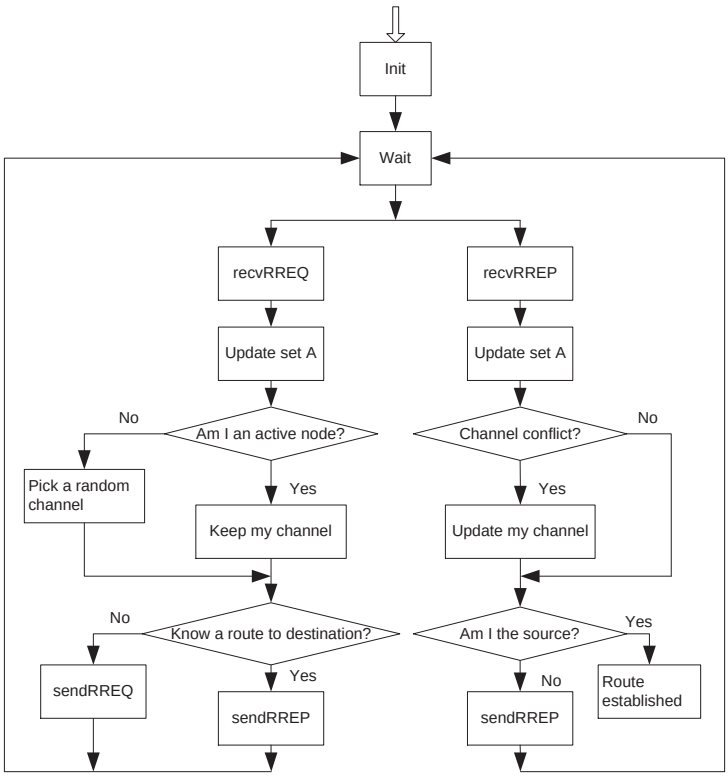


Figure 7: Algorithm flowchart of CA-AODV [48]. This figure shows that if the node does not have a pre-assigned channel, the channel will be chosen randomly from the list of available channels.

at each node based on probabilites for the next hop. Data packets are forwarded deterministically by selecting the highest probability at every node on the path. Thus, a global route is created by using local hop-by-hop information.

Forwarding Dilemma Game (FDG) [85]: Naserian and Tepe propose a game theoretic approach to forwarding flooding packets in MANET with AODV as the underlying routing protocol. The game is played within the network only when a node receives a `HELLO` or any other flooding message since the nodes are the players. The game, called the *forwarding dilemma game (FDG)*, is composed of the number of players receiving the packet, the forwarding cost and the network gain factor and it offers primarily two strategies - forwarding or dropping the packet. Using a mixed strategy Nash equilibrium, the probability of forwarding the flooding messages are calculated. The FDG is implemented in AODV with the existing `HELLO` messages used in neighbor discovery. Since the `HELLO` messages are forwarded only to the winners of the game, the number of nodes participating in the route discovery process is reduced. The structures of the `RREQ` and `RREP` packets of AODV have been modified for the calculation of the probability of packet forwarding.

Long Lifetime Route (LLR) [28]: Cheng and Heinzelman argue that many routes in ad hoc networks are short lived, triggering frequent route discovery processes, which in turn account for extra control overhead and packet latency. They propose two techniques which allow the network to select long lifetime routes (LLR).

The g-LLR approach is a global approach where the optimal LLRs are computed in a centralized manner. As such information is not normally available to the nodes in a network, its importance is mostly as benchmark. The d-LLR approach selects long lifetime routes in a distributed manner, using only local information. The authors show that the performance of d-LLR closely matches that of g-LLR.

From a practical implementation point of view, the LLR approach can be used to enhance most existing ad hoc routing algorithms. In addition, having longer lifetime routes at the network layer, also improves the performance of the transport layer protocols built upon them: UDP, TCP as well as

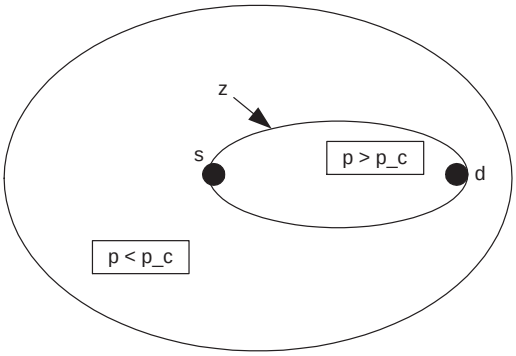


Figure 8: A sketch of the partition of the node of a network. The forwarding probability for the nodes belonging to Z is higher than the critical value p_c. The nodes outside Z forward a packet with probability lower than p_c. [12].

wireless-specific variations on TCP such as TCP-NB (no-backoff).

Polarized gossip protocol for path discovery [12]: Beraldi looks at gossip protocols for path discovery, where a node forwards a packet with some predetermined probability. In contrast to classical gossip algorithms which forward each message with the same probability, this work considers the probability dependent on node locations and distances between each other. We have the polarizing node n with two gossiping probabilities: p_F (the forward probability) and p_B (the backward probability). The message is forwarded with probabilty p_F when the node receives this message from at least one node which is farther from the destination node than the node itself. Otherwise, it forwards the message with probability p_B (see Figure 8). Distances between nodes can be found by simple estimates of relative positions between neighbor nodes which can be determined by using periodic beaconing schemes.

On-demand packet forwarding scheme (OD-PFS) [4]: Al-Karaki and Kamal propose a clustering approach followed by a routing protocol exploiting the clustering framework in MANETs. A fixed and scalable virtual wireless backbone, called the virtual grid architecture (VGA), is created. The physical network topology is mapped onto a virtual grid topology. The routing is then carried out using a combination of hierarchical and virtual backbone routing. Nodes are divided into fixed clusters (also called zones) which require few virtual topology updates. The authors show that the virtual topology is stable as long as at least one mobile node is present in each cluster. With clusters being fixed, the architecture becomes simple and scalable. Network zoning is accomplished by dividing the network into disjoint but adjacent regular shape zones. Zone lengths in homogeneous MANETs are chosen such that two mobile nodes in adjacent zones can always communicate directly with each other. In heterogeneous MANETs, network topology can be changed by varying the nodes transmission range. Clusterheads (CHs) are selected either by periodic elections using an eligibility factor (EF) or dynamically when needed. CHs can also be elected on-demand. A simple *on-demand packet forwarding scheme (OD-PFS)* is implemented over the VGA. Four standard directions (North, South, East and West) are used for simple packet forwarding using the well-known route request (`RREQ`) and route reply (`RREP`) cycle. The path includes the direction of the packet to be forwarded. For example, N-W-W-S-W-N-N-N-E is a sample path. A transitive closure approach is used where each CH maintains an adjacency matrix of neighboring CHs with alive links. By calculating successive transitive closures of the adjacency matrix, path existence between a pair of nodes can be computed. The local path restoration in VGA is shown in Figure 9.

QoS routing with traffic distribution (QMRB) [52]: Ivascu et al. use a mobile routing backbone to support QoS in a MANET. The mobile routing backbone (MRB) dynamically distributes traffic within the network and selects the route with the best QoS between a source-destination pair. The proposed scheme classifies the nodes in the network into QoS routing nodes (QRN), simple routing nodes (SRN) or transceiver nodes (TN). QRNs possess QoS guarantees, SRNs simply route packets through the network while TNs send and receive packets but cannot relay them. The MRB is formed by these different types of nodes while it is not essential that all nodes in the network join the MRB. Nodes not joining the MRB may still communicate with it through a working link.

Node classification for the MRB is computed by the four QoS support metrics (QSMs) for each pair

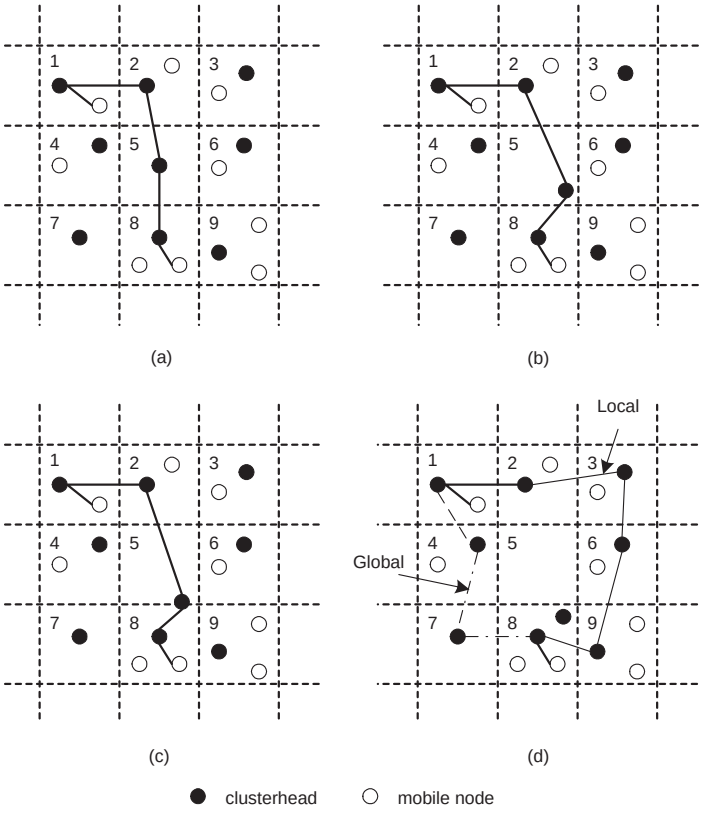


Figure 9: The illustration of local path restoration in VGA [4].

of nodes.

- Static resources capacity (SRC): computed by the weighted sum of the size of the node packet queues, speed of the CPU, power of the battery and the maximum available bandwidth.
- Dynamic resources availability (DRA): indicates the current load in the resource usage of a node. The usage rate of the static resources are used to calculate the available dynamic resources.
- Neighborhood quality (NQ): the number of nodes in the neighborhood of a node which can successfully forward packets.
- Link quality and stability (LQS): the power of signal received and the statistical stability of its links.

The node aptitude is computed based on the node classification by following the formula:

$$MN_{aptitude} = \mu SRC + \eta DRA + \sigma NQ + \omega LQS + \phi BW$$

where μ, η, σ, ω and ϕ are coefficients and BW is the available bandwidth. Once the MRB is set up, route discovery is initiated using `RREQs` along the MRB.

Adjusted probabilistic route discovery [1]: Abdulai et al. observe that rebroadcasting route request packets in a MANET leads to extensive control overhead and high levels of channel contention. This work proposes two probabilistic methods aiming to reduce the number of `RREQ` packets using a *predetermined fixed-value forwarding probability*. Unlike other similar algorithms, the proposed mechanism does not use GPS based devices for location tracking but mainly relies on basic topology information. It also uses the minimum connected dominating set (MCDS) requiring global topological information of a network. In case of receiving duplicate packets at a node, the forwarding probability is adjusted. The first probabilistic route discovery scheme is called the *2P-Scheme* in which the nodes are

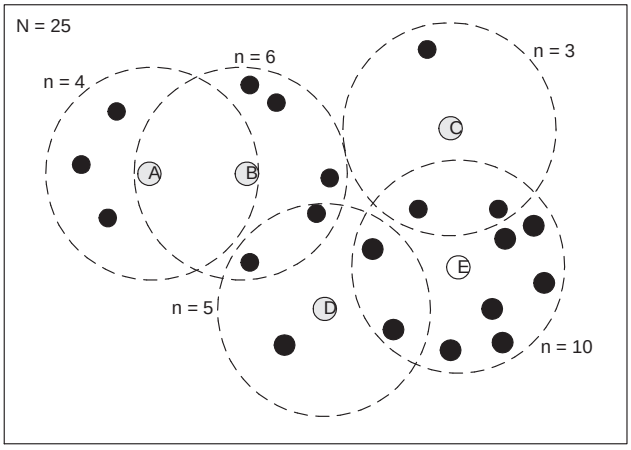


Figure 10: Illustration of two logical groupings of 25 nodes located in sparse and dense regions of a network for the 2P-Scheme algorithm [1].

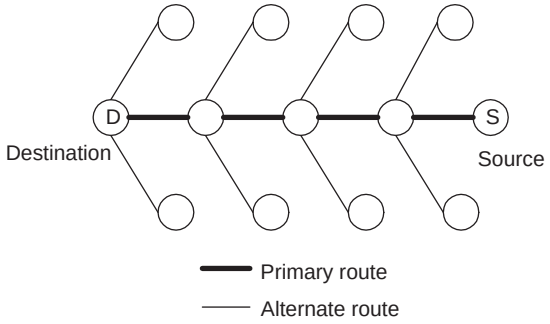


Figure 11: A fish bone structure formed by the primary route and alternate routes [65].

categorized into two groups based on their current neighborhood information. If the node is situated in a sparse region, it gets assigned to Group-1 and if in a dense region, it is assigned to Group-2 (see Figure 10). Nodes in Group-1 are allocated a higher forwarding probability than those in Group-2. In the second scheme, *3P-Scheme*, the nodes are classified into three groups: Group-1 for sparse regions, Group-2 for medium dense regions and Group-3 for dense regions. For this scheme, the forwarding probabilities are assigned in non-decreasing order.

Adaptive backup routing (AODV-ABR) [65]: Lai et al. provide an extension to the AODV-BR scheme which used the concept of backup routes to AODV. It sets up a mesh and multipath routing using `RREP` messages and aims to reduce control overhead. The mesh structure is created by overhearing data packets transmitted from the nodes in the neighborhood (see Figure 11). This helps in reducing control messages and also react faster to the topology changes. Whenever a link break is detected at a node, the node itself initiates a handshake procedure with its neighbors and repair the broken route. Backup route request (`BRRQ`) and backup route reply (`BRRP`) messages are used in the route repair process. The authors also suggest combining AODV-ABR with local route repair once the route breaks by broadcasting a `RREQ` message from the node with the broken link.

Low overhead dynamic route repairing [135]: Yu et al. repair broken routes dynamically by using information from overhearing the nodes. Once the route is down, the proposed protocol intelligently replaces the failed links with backup links along the main route. An initial main route is constructed between a source and destination pair along which the data packets are transmitted. While these packets are forwarded, neighboring nodes overhear these transmissions. These nodes are actually potential candidates for replacing a node on the main route in case of a failure. These candidate nodes are recorded in the packet headers and used during the route reconstruction in the future. The route

construction uses a main route request (MREQ) packet initially, similar to a RREQ packet in AODV. The MREQ is flooded throughout the network and once it arrives at the destination, the main route reply (MRRP) is sent back. The MRRP keeps track of the hop count between the source and destination. This hop count is used during route repair when a particular link on the main route goes down. A repair query (REPQ) packet with the hop count of that node is forwarded in case of the main route failure. A node receiving the REPQ checks to see if it has a saved route to the destination. If so, it sends a repair reply (REPR); otherwise, the hop count value is compared with its own hop count. If its hop count is smaller, it propagates the REPQ packet towards the destination, else the packet is dropped. This enables nodes closer to the destination to receive the REPQ and reply with a valid backup route.

Link availability-based QoS-aware (LBAQ) routing [136]: Yu et al. propose the LBAQ routing protocol based on node mobility prediction and link quality measurement. While a node moves, it may experience varying capacity, reliability and bandwidth availability. Instead of trying to predict the mobility patterns of the mobile links, the link availability is incorporated into the routing metrics to help choose the link with the highest availability in the route. The expected transmission count (ETX) along with power consumption is also used as a route metric. The route metrics used are:

- Link availability: probability that two nodes of a link stay directly connected at a time $t_0 + t$ provided that they were connected at time t_0.
- Link quality: the number of retransmissions required to send a packet on a link between two nodes (ETX). This is computed by measuring the loss rate of broadcast packets between the node pairs.
- Energy consumption: power consumption at each hop.

Using the metrics above, a combined cost function is designed based on which the route is constructed.

$$D_i(T_i) = \gamma_0(E_i) + \gamma_1(Q_i) + \gamma_2(1 - L_i(t))$$

where γ_0,γ_1 and $\gamma_2 = 1 - \gamma_0 - \gamma_1$ are the weighting coefficients, L_i and Q_i are availability and quality of link i, E_i is the energy consumed on link i, T_i is the traffic carried on path i, and D_i is the final cost of the route. A source node sends a RREQ packet during route discovery. When an intermediate node receives this RREQ packet, it calculates its own cost from the equation above and forwards the packet with this new information. The total cost to the destination is computed when the RREQ reaches the destination node. At the destination, the node waits for a fixed number of RREQs before selecting the route with the least cost. If a link breaks, then an alternate route is selected for data transmission.

Labeled successor routing (LSR) [101]: Rangarajan and Garcia-Luna-Aceves notice that many modern on-demand protocols are built on top of AODV, using the same destination sequence numbers. Thus, they inherit the performance problems of AODV: (a) most route requests are answered by the destination and (b) it can suffer from temporary loops, de-facto partition and count-to-infinity. The LSR approach is an attempt to overcome these problems by using the information already needed in route requests to establish and maintain loop-free routes and allows other nodes than the destination to initiate route replies. LSR uses unique source-sequenced labels (SSLs) of flooded route request (RREQ) messages to build loop free paths within the network. Directed acyclic graphs (DAGs) from source to destination are created and route replies are forwarded to the source through the reverse paths. In LSR, the destination must answer every RREQ it receives since every node relaying a RREQ associates a relay sequenced label (RSL) with the SSL of the forwarded RREQ. The RREPs take different paths along the DAG built by the RREQs. There could be multiple DAGs through the same node due to different RREQs forwarded through the node. To avoid path loops, nodes use the RSL to select only a unique RREP with an SSL and drop the others. The LSR creates unique route request labels (RRLs) derived from the SSL of the RREQ forwarded. A source floods a RREQ SSL and the participating nodes on this DAG store the SSL as the RRL for that particular destination. The source can then forward packets through any of these nodes towards the destination.

Stable weight based on demand routing protocol (SWORP) [123]: Wang et al. propose a weight based mechanism for routing in MANETs. Weights are assigned to different routes during route discovery using the route expiration time (RET), the error count (EC) and the hop count (HC). Route discovery in SWORP is similar to DSR with a source node initiating a RREQ message. The destination node sends

the RREP when it receives an RREQ for itself. When multiple RREQs are received from different paths, the destination node calculates the RET, the EC and the HC for each path. With these metrics, the weight values for each path is calculated by the destination node and the route with the largest value is chosen as the primary route. The RET is computed by the destination node using the GPS signal strength to determine when the link between a pair of moving nodes may be disconnected. The EC denotes the number of link failures caused by a mobile node while the hop count is the distance from the node in terms of hops. Using these values, the weight function calculates the weight for each path using the following equation:

$$W_i = C_1 \cdot \frac{RET_i}{MaxRET} + C_2 \cdot \frac{EC_i}{MaxEC} + C_3 \cdot \frac{HC_i}{MaxHC}$$

where C_1, C_2 and C_3 are the weighing factors with their sum equal to one. Other important routing protocol functions such as route maintenance is carried out similar to DSR and AODV with a RERR packet generated when a link is broken.

Recycled path routing (RPR) [38]: Eisbrener et al. present a new strategy towards broadcasting route request (RREQ) packets in MANETs during route discovery. It uses expired routes stored in the route cache to make an educated decision on forwarding RREQ packets towards the destination. The authors implement controlled flooding in the direction of the destination node but without any prior location information. RPR propagates RREQs without flooding every node and uses the *hot* or *cold* concept. Nodes closer to the destination are considered hotter than the nodes farther away. These hot or cold values are based on the route request for a particular destination node. Hot nodes rebroadcast the RREQ message while cold nodes discard it. The reasoning behind this forwarding strategy is the fact that a broken or expired route can be used towards finding the new route. Essentially, expired route caches are reused unlike other on demand protocols. If the destination node is within the expired route cache of a node receiving the RREQ, that node is also considered hot. On the other hand, if the node is colder than a pre-defined threshold, the RREQ is not forwarded at all. Initially, the nodes have empty route caches, meaning no hot or cold settings. During the network initiation, all nodes broadcast packets using traditional flooding and setup the routes. Timers are used to maintain the expired routes. Once this timer expires, the expired routes are purged from the route cache to free up memory and remove redundancy.

Gathering based routing protocol (GRP) [3]: Ahn presents the gathering based routing protocol which collects network information during route discovery to be used later by the source node. Initially, the source node broadcasts a destination query (DQ) packet which is continuously forwarded towards the destination. This process is similar to the route discovery process in DSR or AODV using RREQs. When the DQ reaches the destination node, a network information gathering (NIG) packet is forwarded by the destination. This NIG packet is then propagated over the entire network collecting information. Every NIG packet is assigned a sequence number. Nodes receiving a NIG packet with a new sequence number attach network information to this packet and forward it along effective outgoing links (EOLs). EOLs are those links in which the NIG packets are not received. Once the NIG packet arrives at the source node, the optimal path is calculated by the source node based on the collected network information. Data packets are then transmitted through this calculated route.

Source routing with local recovery (SLR) [108]: Sengul and Kravets start from the observation that although on-demand routing reduces the cost of routing in high mobility environments, the route discovery process, which is typically done through network-wide flooding, consumes a significant amount of bandwidth. This is especially expensive if the route discovery must be repeated due to links broken due to node mobility. To alleviate this problem, the authors propose *bypass routing*, a process which patches a route using local information acquired on-demand, without the need of network-wide flooding. The SLR protocol is an implementation example of the bypass routing approach.

Bypass routing in SLR localizes the reaction to a route failure and initiates only a local recovery procedure. Using link state information from the MAC layer, a local patch on the route is created bypassing the broken link. By using local recovery with link state information, the chances of recovering from a broken route increases. When a route fails, first the local route cache is searched for an alternate route. If no route is available, bypass recovery is initiated by querying the neighbors. Based on replies from these nodes, the path is again reconstructed. MAC caches provide the connectivity information

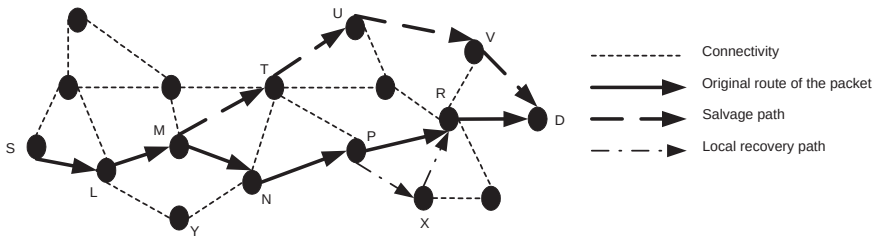


Figure 12: Error recovery example [108].

to immediate one-hop neighbors and the state is updated by each node whenever any communication is heard in the neighborhood. MAC caches also help determine the validity of the routes in the route caches. Error recovery is initiated by salvaging used route caches, bypass recovery and error reporting (see Figure 12). The node looks for alternative routes to the destination in its cache to salvage a packet. If the salvaged packet reaches its destination, the source receives an enhanced route error message from the destination stating that the salvaged route is in fact an alternate to the broken route.

Hint based probabilistic protocol [13]: Beraldi et al. propose a probabilistic forwarding framework which uses meta-information to forward packets towards the general direction of the destination. The meta-information is provided in terms of *hints* at each node. A hint, h_{id}, computed by node i with respect to the destination node d, is represented by a positive value indicating the *chance* of node i residing in the neighborhood of node d. A lower hint value equates to a higher probability with $h_{id} = 0$ when nodes i and d are one-hop neighbors.

When a node forwards a packet to the destination d it will try sending to the neighbor with the best hint which has not been tried before for the same packet (effectively trying various alternatives in the order of the hint). If no such neighbor exist, the packet is discarded. All the nodes broadcast periodic heartbeat packets which are then used to create a vector of time information for each neighbor. This time information vector is utilized to calculate the hint values. Hints are disseminated by broadcasting the control messages through the periodic beacon messages. Every node receives hints from nodes at most L hops around it. This is the *lookahead* value for the protocol. The authors show that a small lookahead value is sufficient for a node to gather correct hints. Smaller lookahead values also have an added advantage of lower control overheads.

Labeled distance routing (LDR) [40]: LDR, by Garcia-Luna-Aceves et al., is based on AODV but uses distance labels instead of sequence numbers to ensure loop freedom in the network. It utilizes a loop free invariant for each destination with the sequence numbers which can only be incremented by the destinations. The sequence numbers are used for path resets. An *advertisement* denotes an offer for a route to the destination while a *solicitation* denotes a request for information to a destination. To ensure loop freedom, `RREQs` are disseminated within a tree which enforces a strict ordering of feasible distances along successor paths. The tree is created by the regular reverse-path flooding as in AODV. Ordering of feasible distances from source to destination is attained by adhering to the following constraints: i) numbered distance condition, ii) feasible distance condition, and iii) start distance condition. Since LDR is dynamic, the computed path graph must also dynamically adapt to changing node positions and link conditions. A route discovery is initiated by a source node with a solicitation for a destination. Nodes relaying the solicitation participate in the route computation by caching the path data for a fixed period of time. The destinations or nodes having a path to the destination then send the `RREPs` along the reverse path.

Dynamic backup routes routing protocol (DBR2P) [125]: Wang and Chao present an on-demand routing protocol which does not require any routing table. Destination nodes send back entire routes to the source node while setting up multiple backup routes dynamically. These backup routes are used in the event of a link failure. Intermediate nodes also receive and transmit request packets from the source nodes to gather more information in order to create the backup routes. During route discovery, the source sends an `RD-request` packet with a unique sequence number and a route content field storing

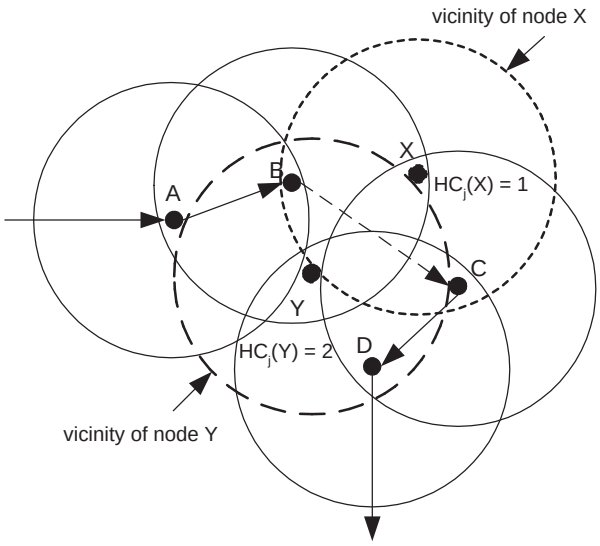


Figure 13: Vicinity of node [75].

the addresses of all the nodes in the path. The sequence number distinguishes between route discoveries from different source and destination pairs. `RD-request` packets received with duplicate sequence numbers are discarded. Route maintenance uses acknowledgment packets at the data link layer to detect lost or corrupted packets. Passive acknowledgments, where a node hears the packet forwarded by its neighbor to the next node, can also be used by the protocol. On link failure, data packets are buffered at the failed node till a backup route comes from an upstream node which caches backup routes. These backup routes are saved at specific backup nodes on the route during path discovery.

Refinement based routing (RBR) [75]: Liu and Lin propose a refinement based route maintenance mechanism which adds proactive route selection and maintenance to on-demand routing approaches. RBR consists of two mechanisms: passive probe route-redirection ($P - PR_2$) and active probe route-redirection ($A - PR_2$). $P - PR_2$ dynamically repairs broken routes through a node called a passive-redirector that redirects the broken source to itself before connecting it to a new node located in the vicinity of the redirector and close to the broken path. $A - PR_2$ uses an active redirector which searches for available shorter paths progressively by overhearing two nodes on the original path. These nodes should be two-hops away and within the vicinity of the active-redirector. These two mechanisms are incorporated into AODV routing protocol. Figure 13 illustrates the concept of the vicinity of a node.

3.1.1. Comparison

Source routing protocols have been a research topic of high interest for the ad hoc community. There are several reasons for this. First, while table driven protocols can be perceived as extensions or adaptations of existing protocols from the wired domain, reactive routing is specific to ad hoc networks. Furthermore, reactive routing is best adapted to the most challenging incarnations of the ad hoc networks: situations where the mobility of the nodes is so high that any route found between a source and a destination will inevitably be temporary in nature. Not only that routes are created on-demand, but the probability of a route failing is so high that the efficiency of responding to failures is a major design consideration.

The baseline for this class of protocols is set by AODV and DSR, both of them have several independent implementations for various operating systems. A large number of other protocols in this class can be seen as attempts to improve the performance of these protocols under various operating scenarios. Some of these scenarios involve additional assumptions about the nodes capabilities, for instance the ability to possess a way to accurately measure location, such as a GPS, or the ability to measure the strength of a received signal.

One of the main improvement directions have been about the handling of recovery of failed routes. This can be approached from several directions:

- Improving the speed of rebuilding the routes: protocols of this class are TORA [89], AODV-ABR[65] (rebuilding with backup routes), low overhead dynamic route repair[135], SLR[108] (using bypass routing), DBR2P[125] (backup routes), RBR[75] (using route redirectors)
- Choosing paths which are predicted to be more stable: ABR[116] (based on associativity), SSBR[37] (based on signal stability), FORP[113] (by predicting future changes), LLR[28] (favoring long lifetime routes), QMRB[52] (???) and LBAQ[136] (based on prediction of future link availability).
- Based on prediction of failure and preemptive rebuilding of routes: the preemptive routing by Goff et al. [47], SWORP[123] (using a GPS signal to predict when a pair of moving nodes might disconnect).

Another direction of optimization is the lowering of the cost of route discovery. One immediate way to perform this is by taking advantage of location information. This way, these protocols represent a bridge towards location-aware routing. From the protocols discussed in this section, polarized gossip [12] and OD-PFS[4] fall in this category. Other protocols work to lower the cost of route discovery without relying on location: Adjusted probabilistic route discovery [1] and recycled path routing (RPR) [38].

Another direction of work is to improve upon the transitory behavior of the baseline protocols - for instance LSR[101] and LDR[40] improve upon AODV's behavior with regards to temporary loops and the phenomena of count-to-infinity.

Finally, there are some protocols which extend upon the baseline by considering various additional networking challenges, such as Quality of Service (LBAQ[136] and QMRB[52]), interference and channel assignment [48] or intermittent connectivity (SCaTR[17]).

The novel nature of source originated routing encouraged researchers to branch out from the traditional comfort zone of networking (which implies reliance on graph theory and flow optimization as the mathematical background). A number of innovative and interdisciplinary approaches have been applied to source originated routing. In the papers surveyed in this section, we have two approaches based on ant colony optimization (ARA[50] and DAR[104]), an approach based on game theory FDG[85] as well as a probabilistic routing approach([13]).

Table 3.1.1 summarizes the papers reviewed in this section and compares some of their key features.

3.2. Table-driven protocols

Table driven protocols always maintain up-to-date information of routes from each node to every other node in the network. Routing information is stored in the routing table of each node 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, all of them have 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) [93]: Perkins and Bhagwat introduced Destination-Sequenced Distance-Vector (DSDV), one of the earliest ad hoc routing protocols. As many distance-vector routing protocols, it relies on the Bellman-Ford algorithm. Every mobile node maintains a routing table which contains the possible destinations in the network together with their distance in hop counts. Each entry also stores a sequence number which is assigned by the destination. Sequence numbers are used in the identification of stale entries and the avoidance of loops. In order to maintain routing table consistency, routing updates are periodically forwarded 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 (must fit in a single packet) and are used to transmit those entries from the routing table which have changed since the last full dump update. When a network is stable, incremental updates are forwarded and full dump are usually infrequent. On the other hand, full dumps will be more frequent in a fast moving network. In addition to the routing table information, each route update packet contains a distinct sequence number assigned by the transmitter. The route labeled with the most recent (highest number) sequence number is used. The shortest route is chosen if any two routes

Table 1: Reactive routing protocols comparison

Protocol	**MR**	**Route metric**	**Route repository**	**Route rebuilding**	**CO**
DSR	Yes	SP or next available	RC	New route and notify source	High
AODV	No	Newest route and SP	RT	Same as DSR or local repair	High
TORA	Yes	SP or next available	RT	Reverse link and repair route	High
ABR	No	Strongest associativity	RT	Local broadcast	Medium
SSBR	No	Strongestquit signal strength	RT	New route and notify source	Medium
Goff et. al	Yes	SP	RT	Path discovery before probable route failure	Medium
AQOR	No	Link bandwidth	RT	Initiated from destination	Medium
ARA	Yes	SP	RT	Alternate route or backtrack	Medium
ROAM	Yes	SP	RT	Erase route, start new search	Low
FORP	No	First created route	RT at clusterhead	New route and notify source	Medium
SCaTR	Yes	Proxy contact value	RT	Proxy requests	High
DAR	Yes	Weighted probabilities	Stochastic RT	New route by sending forward ants	Medium
Beraldi	No	Forwarding probability	–	Broadcast	High
QMRB	No	QoS metrics	RT	Same as DSR or local repair	High
Yu et al.	No	SP or backup route	RC	Local backup	Medium
LABQ	Yes	LA, Q, and E	RC	Same as DSR or local repair	High
LSR	Yes	Relay sequenced label (RRL)	RT	Same as DSR or local repair	High
OD-PFS	No	VBR and CH	RT	Local repair	Medium
SWORP	No	RET, EC, HC	RT	Same as DSR or local repair	High
RPR	No	Newest route and SP	RT	Local repair	High
GRP	Yes	SP	RC	Backup route	High
SLR	Yes	SP or next available	RC	New route and notify source	High
Beraldi et al.	Yes	Hint value	RC	Local broadcast	High
LDR	No	Newest route and SP	RT	Same as DSR or local repair	High
DBR2P	No	SP	None	Local repair	Low

MR = Multiple Routes

Route metric: SP = Shortest path; LA = Link availability; Q = Quality; E = energy used; RET = Route expiration time; EC = Error count; HC = Hop count; VBR = Virtual backbone routing; CH = clusterhead

Route repository: RC = Route cache; RT = Routing table

CO = Communication Overhead

have the same sequence number.

Analysis of a Randomized Congestion Control Scheme with DSDV Routing in Ad hoc Wireless Networks [20]: Boukerche et al. describe a randomized version of the DSDV protocol (R-DSDV) where the control messages are propagated based on a routing probability distribution. Local nodes can tune their parameters to the traffic and route the traffic through other routes with lighter load. This implies implementing a congestion control scheme 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 forwarded. 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 send a regular message with a probability $1 - Pr_{n,adv}$ or an update message with probability $Pr_{n,adv}$. The sending rate of the table advertisement 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. The scheme allows the piggybacking of the routing table updates on the regular messages.

Optimized Link State Routing (OLSR) [31]: Clausen et al. designed the OLSR algorithm which improves on the classical link state protocols through several optimizations targeted towards wireless ad hoc networks. These optimizations are centered on specially selected nodes called *multipoint relays* (MPR). First, only MPR's forward messages during the route information flooding process, substantially reducing the total number of messages forwarded. In addition, link state information is generated only by the MPRs, further reducing the amount of data which needs to be disseminated. Finally, the MPRs might choose to report only links between themselves and their MPR selectors. This last technique of partial link state information is a departure from the customary approach of link state protocols which relay on the dissemination of the full link state.

The multipoint relay aims 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. The multipoint relay set must be chosen such 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 neighbors of N has a bi-directional link with the nodes in the MPR set of `N`. These bi-directional links can be determined by periodic `HELLO` packets containing information about all neighbors and their link status. Thus, a route is a sequence of hops from a source to a destination through multipoint relays within the network. The source does not know the complete routes only next hop information to forward the messages.

A Hierarchical Proactive Routing Mechanism for Mobile Ad hoc Networks (HOLSR) [119]: Villasenor-Gonzalez et al. networks where some nodes have significantly higher resources (transmission range, bandwidth, directional antenna and so on). The authors notice that traditional, flat routing protocols can not efficiently exploit the capabilities of the nodes with high resources. For this scenario, the authors propose the HOLSR algorithm which builds upon the OLSR protocol by introducing a hierarchical architecture with multiple ad hoc networks at distinct logical levels within the network.

The HOLSR network arranges the nodes in three topology levels, depending on their capabilities. The low capability nodes at topology level 1 have only one wireless network interface and communicate with nearby nodes. Nodes on topology level 2 are assumed to have two wireless interfaces, possibly relying on different wireless standards, allowing them to communicate with nodes at topology levels 1 and 2. Finally, the nodes at topology level 3 are the most powerful (e.g. airborne nodes) and can have up to three wireless interfaces, allowing them to communicate with nodes at topology levels 1, 2 and 3. At each level, the mobile nodes can self-organize into clusters, with each node participating in multiple topology levels automatically becoming a clusterhead at the lower level.

Each cluster node maintains a routing table with routing information about nodes in the cluster. Higher clusterheads contain bigger routing tables since they have to maintain routes to all nodes at lower clusters. For lower level nodes, this overhead is minimal. A hierarchical topology control (`HTC`) message is used to send cluster membership information from a lower to higher level. Topology control is carried out using these control messages using the clusterheads as the *gateway* nodes.

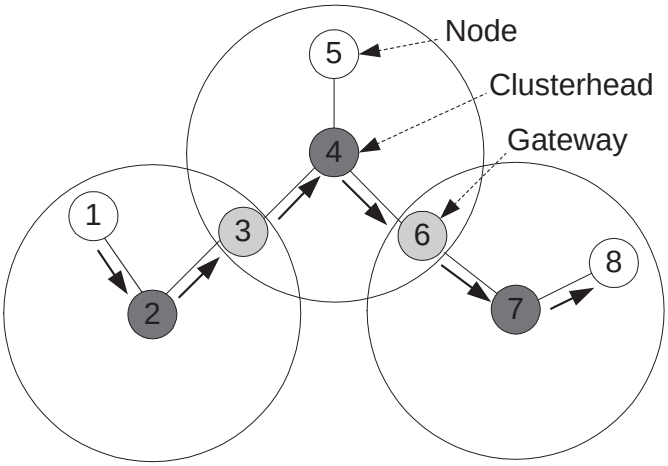


Figure 14: Cluster Gateway Switch Routing [23].

Clusterhead Gateway Switch Routing (CGSR) [23]: The CGSR protocol, by Chiang et al., uses a distributed algorithm called the Least Cluster Change (LCC). By aggregating nodes into clusters controlled by the clusterheads, a framework is created for developing additional features for channel access, bandwidth allocation and routing. Nodes communicate with the clusterhead which in turn communicates with other clusterheads within the network (see Figure 14).

The selection process of a clusterhead is an important task since changing clusterheads frequently adversely affect the resource allocation algorithms. Thus, cluster stability is of primary importance in this scheme. The LCC algorithm is considered stable since the clusterheads 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. 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 members of more than one cluster; therefore, they need to communicate using different CDMA codes based on their respective clusterheads. The main factors affecting routing in these networks are token passing (in clusterheads) and code scheduling (in gateways). A packet is routed through a collection of these clusterheads and gateways in this protocol.

Wireless Routing Protocol (WRP) [83]: Murthy and Garcia-Luna-Aceves propose WRP which builds upon the distributed Bellman-Ford algorithm. 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 must be forwarded to all the neighbors of a node and contain all the routes in which the node is aware of. Since these are just update messages, only the recent path changes are included instead of the whole routing table. To keep the links updated, empty `HELLO` packets are forwarded at periodic intervals only if no other update messages need forwarding.

Global State Routing (GSR) [27]: Chen and Gerla propose the GSR protocol, where the control packet size is adjusted to optimize the MAC throughput.

Each node maintains the neighbor list and three routing tables containing the topology, the next hop, and the distance respectively. The neighbor list contains all neighbors of the current node. The topology table contains the link state information and a timestamp indicating the time in which the link state information is generated. The next hop table contains a list of next hop neighbors to forward the packets while the distance table maintains the shortest distance to and from the node to various destinations. A weight function computes the distance of a link which may be replaced by other QoS routing parameter.

Initially each node in the network starts with an empty neighbor list and a topology table. It

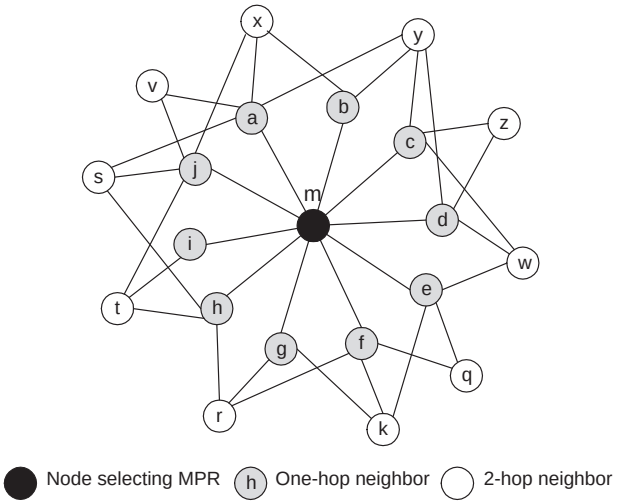


Figure 15: Example for multipoint relay selection [82].

learns about its neighbors by the sender field of the incoming packets. By processing these packets to obtain link state information, the best route to the destination is computed. After all the routing messages are processed, the routing table is created and shared with other nodes by broadcasting it to the next hop neighbors. This process is carried out periodically to maintain the most up-to-date information.

Source-Tree Adaptive Routing (STAR) [43]: Garcia-Luna-Aceves and Spohn propose STAR where each node maintains a source tree which contains preferred links to all possible destinations. Nearby source 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. The link state update 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. The 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 [105].

OLSR with Quality of Service (QOLSR) [82]: Munaretto and Fonseca design the QOLSR protocol by adding the QoS parameters of delay and bandwidth to the standard OLSR. Three new heuristics, QOLSR1, QOLSR2 and QOLSR3, are proposed for multipoint relay selection (see Figure 15). These heuristics select multipoint relays (MPRs) within the network based on various QoS parameters. QOLSR1 chooses the neighbor node which can *reach* the largest number of nodes (node with the maximum degree). The priority is given to the neighbor with smallest delay in the case of multiple neighbors with identical maximum degrees. QOLSR2 prioritizes the neighbor with the smallest delay. If multiple nodes have the same minimum delay, then the node with the highest degree is chosen. The final heuristic selects the node with the smallest delay among neighbors within a two hop distance.

3.2.1. Comparison

In contrast to source initiated routing, table driven routing has extensive precedents in the research done for routing in the wired domain. Nevertheless, the requirements of the ad hoc routing are such that none of the wired routing protocols could be successfully transferred into the ad hoc wireless domain.

On the other hand, the two main classes of wired routing protocols have inspired their own classes of protocols in table driven ad hoc routing.

One of these classes is the *distance vector protocols*, where the nodes maintain only a local topology, and use the distributed Belman-Ford algorithm (or its variations) to maintain the routing tables. In the wired Internet this class contains protocols such as RIP and IGRP. In the ad hoc network domain the

Table 2: Proactive routing protocols comparison

Protocol	Tables	Update interval	Critical node	Routing metric	CO
DSDV	2	Periodic	–	SP	L
R-DSDV	2	Probabilistic	–	SP	L
OLSR	3	Periodic	–	SP	H
HOLSR	3	Periodic	–	SP	H
CGSR	2	Periodic	Clusterhead	SP	L
WRP	4	Periodic	–	SP	L
GSR	3	Periodic only with neighbors	–	SP	L
STAR	1	Only at specific events	–	SP	L
QOLSR	3	Periodic	–	Degree, delay, & HC	H

Routing metric: SP = Shortest path; HC = hop count

CO = Communication Overhead [High = H; M = Medium; L = Low]

most popular protocol of this class is DSDV[93], another protocol being WRP[83]. The other class of protocols is the *link state routing* protocols where the routers exchange full topology information, and then use a graph-theoretic shortest path algorithm (such as Dijkstra's) on the resulting graph. On the wired Internet this class is represented by algorithms such as OSPF and IS-IS. The most representative example in the ad hoc wireless domain is the OLSR[31] protocol. Between these two classes we find protocols which transfer a partial topology, such as STAR[43].

Most of the recent work on table-driven protocols can be seen as improvements on these baseline distance vector and link state approaches.

One direction of research is the adaptation of the routing decisions to the traffic, proposed in the randomized congestion control scheme for DSDV [20]. Another class of protocols aim to improve the scalability of table-driven protocols. The subnetwork approach, successfully applied on the wired Internet, can not be directly applied in ad hoc networks due to the much more variable connection structure. On the other hand, the connections in ad hoc networks are strongly correlated with the physical proximity, which allows the development of clustering approaches (e.g. CGSR[23]), possibly integrated with a hierarchical model (such as in HOLSR [119]). Another direction of research is the consideration of quality of service, as in QOLSR [82].

Table 3.2.1 summarizes the protocols reviewed in this section and compares some of their key features.

3.3. Hybrid protocols

The hybrid routing schemes combine elements of on-demand and table-driven routing protocols. The general idea is that area where the connections change relatively slowly are more amenable to table driven routing while areas with high mobility are more appropriate for source initiated approaches. By appropriately combining these two approaches the system can achieve a higher overall performance.

Zone Routing Protocol (ZRP) [106]: The ZRP protocol, designed by Samar et al. is designed to be used in large scale networks. The 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.

Local neighborhoods, called *zones*, are defined for nodes (see Figure 16). The size of a zone is based on ρ factor defined as the number of hops to the perimeter of the zone. There may be various overlapping zones which helps in route optimization.

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. The Interzone Routing Protocol (IERP) uses a reactive approach for communicating with nodes in different zones. Route queries are forwarded to peripheral nodes using the bordercast resolution protocol (BRP). The ZRP architecture can be seen in Figure 17.

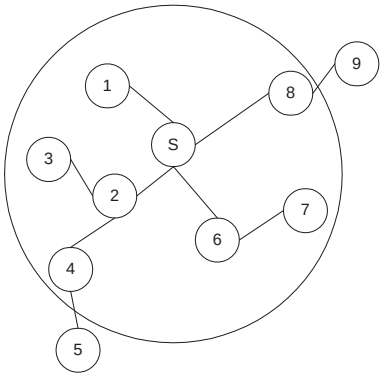


Figure 16: Example routing zone with $\rho = 2$ [106].

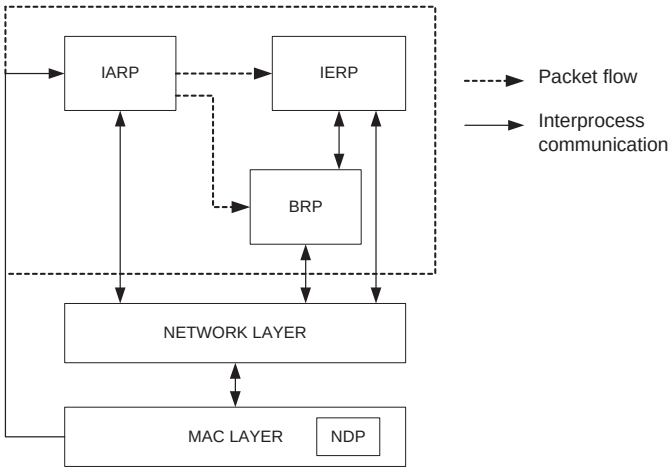


Figure 17: ZRP architecture [106].

Fisheye State Routing (FSR) [90]: Pei et al. propose the FSR protocol which takes inspiration from the "fisheye" technique of graphic information compression proposed by Kleinrock and Stevens. When adapted to a routing table, this technique means that a node maintains accuracy distance and path quality information about its immediate vecinity, but the amount of detail retained decreases with the distance from the node. Each node considers a number of surrounding fish-eye scopes, areas which can be reached with 1, 2, ... hops. A higher frequency of update packets are generated for nodes within smaller scope while the updates are fewer in general for farther away nodes. Each node maintains a local topology map of the shortest paths which is exchanged periodically between the nodes. With an increase in size of the network, a "graded" frequency update plan can be adopted across scopes to minimize the overall overhead.

This approach makes FSR an implicit hierarchical protocol. Its main advantage is the significant reduction of the control overhead.

Landmark Ad hoc Routing (LANMAR) [91]: Pei et al. propose LANMAR which builds subnets of groups of nodes which are likely to move together. A *landmark* node is elected in each subnet, similar to FSR [90]. The LANMAR routing table consist of only the nodes within the scope and landmark nodes. During the packet forwarding process, the destination is checked if it is within the forwarding node's neighbor scope. If so, the packet is directly forwarded to the address in the routing table. If a packet on the other hand is destined to a farther node, it 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. During the link state update process, the 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) [2]: RDMAR, by Aggelou and Tafazolli, has distinct route discovery and route maintenance phase. However, the route discovery broadcast messages are limited by a maximum number of hops calculated using the relative distance between the source and the destination. Each node also maintains a routing table containing the next hop neighbor of each known destination, an estimated relative distance between all known source and destination nodes, a timestamp 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 and the estimated speed of the destination node. Each node also maintains–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 pertaining to the most recent route discovery.

Route discovery and route maintenance is carried out by broadcasting route request packet 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 the route has failed, the intermediate node forwards a *failure notification* to the source node by unicasting it to all neighboring nodes. When a node receives a failure notification, it updates its routing tables accordingly.

Scalable Location Update based Routing Protocol (SLURP) [127]: SLURP, by Woo and Singh, develops 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 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.

Zone based Hierarchical Link State routing protocol (ZHLS) [56]: Joa-Ng and Lu propose ZHLS routing protocol where a hierarchical structure is defined by non-overlapping zones with each node having a node ID and a zone ID. These IDs are calculated using an external location tool such as GPS. The hierarchy is divided into two levels: the *node level topology* and the *zone level topology*. There are no clusterheads in ZHLS.

When a route is required for a destination located in another zone, the source node broadcasts a zone-level location request to all other zones. Once the destination receives the location request, it replies with the path. In this technique, only the node and zone IDs of a node is required to discover a path. There is no need for updates as long as the node stays within its own region and the location update is required only if the node switches regions. The only drawback of ZHLS is that all nodes should have a preprogrammed static zone map to recognize the zones created in the network. This may not be possible in scenarios where the network boundaries are dynamic in nature. On the other hand, it is suitable for the networks deployed with fixed boundary lines.

Distributed Spanning Tree (DST) routing [96]: Radhakrishnan et al. present a routing algorithm which uses distributed spanning trees. There can be regions of different stability in the network and a backbone network must be created within the stable regions. All the nodes in the network are aggregated into a number of trees rooted at a particular node. These trees are composed of *root* and *internal* nodes. The root node can make various decisions such as whether the current tree could join with another tree, whether other nodes could join at appropriate positions within the tree, and so on. The other nodes in the tree are considered regular nodes with no extra or unique functionality. Each node can be in three

different states: router, merge, and configure. DST proposes two strategies to determine a route between a source and a destination pair:

1. Hybrid Tree Flooding (HTF): In this scheme, the source sends the control packets to all the neighbors and adjoining bridges in the spanning tree. Each packet is remained static at these places for a specific *holding time*. This serves as a buffering strategy for these nodes to send the packets as network connectivity increases with time. Hence, the network becomes gradually more stable and lesser number of packets are dropped due to link failures.
2. Distributed Spanning Tree (DST) shuttling: In this approach, the source sends the control packets to the tree edges till each of them reaches a leaf node. When a packet reaches the leaf node, it is forwarded to a shuttling level, i.e. a particular height on the tree. When the packet arrives at the shuttling level, it is sent down the adjoining bridges. This helps to selectively forward the control packets within the network.

The drawback with such an architecture is the existence of a single point of failure for the entire tree. If the root node fails, the entire routing structure falls apart. Also the holding time metric may factor in some additional delays and may not be suitable for transmitting real-time data.

Distributed Dynamic Routing (DDR) Algorithm [87]: Nikaein et al. propose a tree-based routing protocol without the need of a root node. Periodic beacon messages are exchanged among neighboring nodes to construct a strategy tree. These trees within the network form a forest with the created gateway nodes acting as links between the trees in the forest. These gateway nodes are regular nodes belonging to separate trees but within transmission range of each other. A zone naming algorithm is used to assign a specific zone ID to each tree within the network. Hence, the overall network now comprises of a number of overlapping zones (if each tree is considered to be a zone).

The DDR algorithm comprise of the following six phases: (i)preferred neighbor election; (ii)intra-tree clustering; (iii)inter-tree clustering; (iv)forest construction; (v)zone naming; and (vi)zone partitioning. Initially, each node starts with the preferred neighbor election phase in which the preferred neighbor is the one with the highest number of neighbors. A forest is constructed by connecting each node to their preferred neighbor after which the intra-tree clustering scheme is used to generate the zone structure. It also helps set up the intra-zone routing table for communication within the tree itself. Through the inter-tree clustering, connectivity between trees is achieved. The zone naming algorithm is executed to assign zone IDs to each of the zones, resulting of partitioning the network into the various zones based on the assigned zone IDs. This work is extended in [86] by introducing a routing framework which uses the intra and inter zone routing tables created in DDR.

A4LP routing protocol [120, 121]: A4LP, by Wang et al., 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. A4LP 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*. A4LP, is also both *location-* and *power-aware* routing protocol supporting asymmetric links that may be suitable for heterogeneous MANET.

Hybrid ant colony optimization (HOPNET) [122]: Wang et al. present a hybrid routing algorithm based on Ant Colony Optimization (ACO) and zone routing. It considers the scenario of ants hopping from one zone to the next with local proactive route discovery within a zone and reactive communication between zones. The algorithm borrows features from ZRP and DSR protocols and combines it with ACO based schemes. The forward ants are sent only to border nodes. These forward ants are then directed towards the destination node by using the nodes' local routing table. The ants move from one zone to another via border nodes and by using available local routing information. The zone approach achieves the scalability. Link failures are handled within a zone without flooding the network. Inter and intra zone routing tables are always maintained which can efficiently rediscover a new route in case of a link failure. An example scenario is shown in Figure 18.

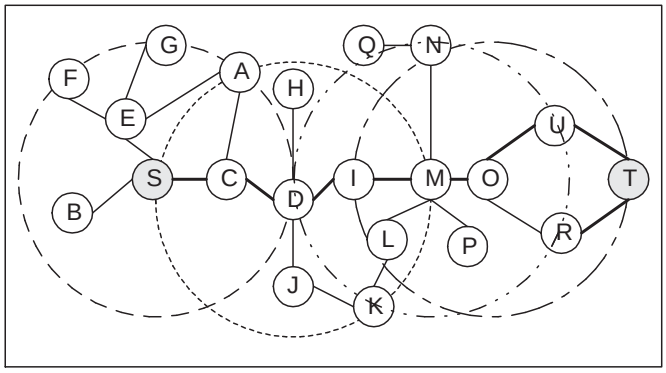


Figure 18: Example scenario [122]. If the radius of the zone is 2, for node S, nodes A, D, F, G are boundary nodes, and nodes B, C, E are interior nodes while all other nodes are considered exterior nodes.

Link reliability based hybrid routing (LRHR) [129]: Xiaochuan et al. observe that frequent topology changes in MANETs may require the dynamic switching of table-driven and on demand routing strategies. The LRHR protocol achieves this switching in a smooth and adaptive fashion. Each node operates in a *promiscuous* receive mode to overhear any packet transmission in the neighborhood and setup multiple routes from the source to the destination. The LRHR assigns edge weights between links based on the link reliability. The higher the value of edge weight, the greater the reliability. It finally selects the route which has the maximum edge weight sum as the main route. The LRHR primarily operates in the table-driven mode and switches to on demand mode when a source either has no route to a destination or when the time interval between a new route discovery and the previous route discovery phase is larger than the minimum route request interval. The route discovery process in LRHR is similar to DSR. The route maintenance operations are carried out based on the edge weights between the nodes.

Fisheye zone routing protocol (FZRP) [133]: Yang and Tseng combine the zone routing protocol with the fisheye state routing mechanism. By using the concept of a fisheye, a multi-level routing zone structure is created where different levels are associated with different link state update rates (see Figure 19). The source node generates a `RREQ` packet which is *bordercast* to nodes till the destination is reached. These packets are forwarded along the border of the zone. The nodes at the periphery of the zones forward the `RREQ` to a different zone if the destination node is not located within the current zone. A *time to live* (TTL) field is used within the forwarding packets to cover the zone in which the update packets are forwarded. The routing table at each node contains entries for nodes within its own zone and for those in an extended zone. An extended zone is a zone located beyond the inner (basic) zone. Extended zone entries are generally not very accurate due to different update frequencies in different zones. Other procedures such as local route repair in case of a broken link are similar to ZRP.

Ad hoc networking with swarm intelligence (ANSI) [97]: Rajagopalan and Shen propose a hybrid routing protocol utilizing swarm intelligence (SI) to select good routes in a network. SI allows self-organizing systems and helps maintain state information about the network. ANSI employs a highly flexible cost function which uses information collected from local ant activity. The protocol takes advantage of the basic principles of ant based routing algorithms which allows the maintenance of multiple routes to a destination. In ANSI, the nodes using proactive routing perform stochastic routing to select the best path, while those performing reactive routing use the extra routes in the event of route failure. The *pheromone* trail concept allows selecting the routes to the destination from every node. Nodes using reactive routing first broadcast a *forward reactive ant* towards the destination. Once the destination receives the forward ant, it replies with a *backward reactive ant* which updates routing tables for all nodes in the path. If route failure occurs at an intermediate node, that node buffers the packets which could not be routed and sends a forward ant initiating route discovery. Nodes which are part of a less dynamic, infrastructured network maintains routes proactively by periodic routing updates using *proactive ants*. Proactive ants are not returned as the reactive ants, rather they help reinforce the path taken by the ant. Local route management is achieved by reinforcement due to movement of

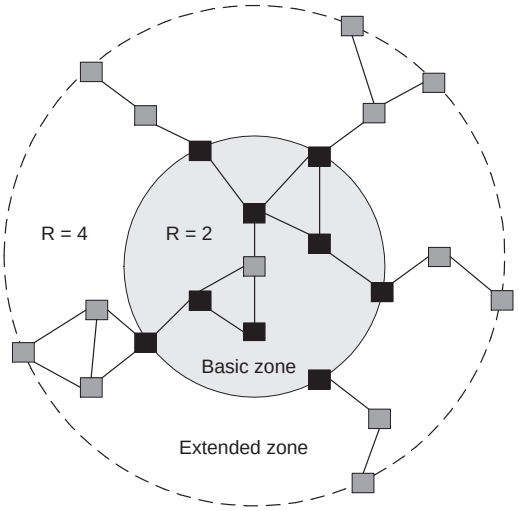


Figure 19: Two level routing zone in FZRP [133].

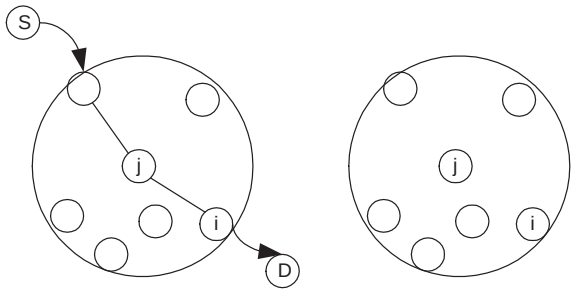


Figure 20: Local reinforcement in ANSI. (a) Reinforcement by data packets. Node i, upon receiving a data packet from S via node j, reinforces the path to node j via j and the source S via j. (b) Reinforcement in neighbor discovery mechanisms. Upon receiving a HELLO beacon from j all nodes i reinforce trails via j. [97].

data packets and an explicit neighbor discovery mechanism (see Figure 20). `HELLO` messages are also broadcast periodically through which network state information is easily exchanged.

Mobility aware protocol synthesis for efficient routing [9]: Bamis et al. propose a new *stability* metric to determine the mobility level of nodes in a network. Using this metric, the nodes can be classified into different mobility classes in which they in turn determine the most suitable routing technique for a particular source-destination pair. Stability uses the concept of associativity, which is the total time for which nodes are connected through beacons. With the level of stability defined, a protocol framework is designed, which operates above the network layer on the protocol stack, determines the optimal routing technique. Minimal changes are needed to the original routing protocols to ensure ease of integration.
Load balancing in MANET shortest path routing [112]: Souihli et al. achieve load balancing to enable efficient routing in MANETs. It has been observed that the load is maximal at the center while it decreases farther from the center of the network. Essentially, the load becomes minimal at the network edges. The authors state that such a load imbalance takes place due to shortest-path routing and propose a new routing metric, the node's *centrality*, when choosing the best route. Thus, instead of selecting the shortest routes, nodes with longer distances from the center of the network are chosen. A new routing metric is proposed as follows:

$$Minimize \frac{1}{n} \sum_{k=1}^{n} \eta(k)$$

where n is the number of nodes in the network and $\eta(k)$ represents the centrality of node k. The $\eta(k)$ value is calculated based on the size of the node's routing table. A greater size denotes closer to the

Table 3: Hybrid routing protocols comparison

Protocol	MR	Route metric	Route repository	Route Rebuilding	CC
ZRP	No	SP	IntraZ and InterZ RTs	Start repair at failure point	M
FSR	No	Scope range	RTs	Notify source	L
LANMAR	No	SP	RTs at landmark	Notify source	M
RDMAR	No	SP	RT	New route and notify source	H
SLURP	Yes	MFR: InterZ, DSR: IntraZ	RC at location	Notify source	H
ZHLS	Yes	SP	IntraZ and InterZ RTs	Location request sent	M
DST	Yes	Tree neighbor forwarding	RTs	Pause times/Shuttling	L
DDR	Yes	Stable routing	IIntraZ and InterZ RTs	Notify source	L
A4LP	Yes	Power consumed	RTs	Notify source	M
HOPNET	No	SP	IntraZ and InterZ RTs	Start repair at failure point	H
LRHR	Yes	Edge weight	RC, RT	Route discovery	H
FZRP	No	SP	IntraZ and InterZ RTs	Start repair at failure point	M
ANSI	Yes	SP	RT	Start repair at failure point	M

MR = Multiple Routes

Route metric: SP = Shortest path; InterZ = Intrazone; IntraZ = Intrazone

Route repository: RC = Route cache; RT = Routing table

CC = Communication Complexity [High = H; M = Medium; L = Low]

center of the network while a lower size indicate otherwise.

3.3.1. Comparison

Hybrid approaches are, in general, justified for large networks - if a network is small, we can usually make a clear decision between source driven or table driven approaches. Similarly, hybrid approaches cover a very wide range of approaches and intellectual ideas. Still we can identify several guiding ideas.

First, there is a group of approaches which performs a differential treatment of the network nodes based on either (a) zones or (b) the nodes participation in a backbone.

The approaches which classify nodes into zones are usually counting the zones from the perspective of the source node. The zones are usually defined based on hop count in protocols such as ZRP[106], FSR[90] and RDMAR[2] although we also have protocols where the zone is based on physical location such as in SLURP[127] and ZHLS[56]. In some cases, the zones might be mobile zones of nodes moving together, as in LANMAR[91].

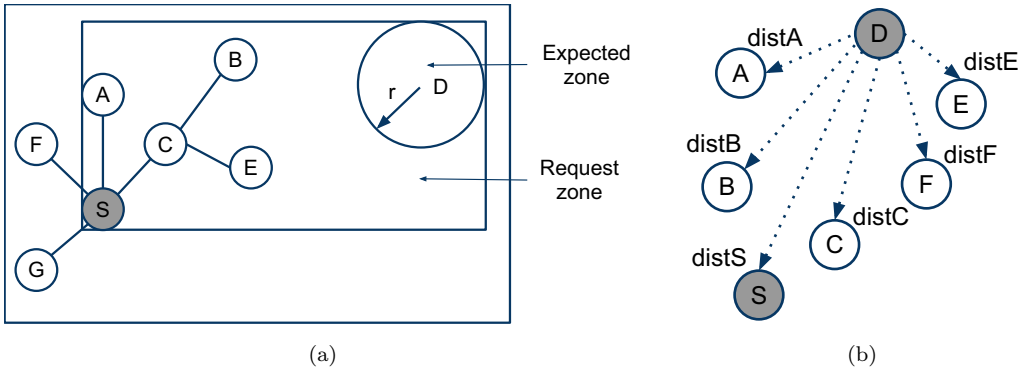
In backbone based approaches a subset of nodes which have more stable connections form a backbone which is frequently organized into a tree structure. The assumption is that nodes in the backbone will use table driven routing, while nodes outside the backbone will be reached with source initiated routing. Example protocols are DST[96] and DDR[87]. The A4LP protocol [120, 121] form a special case as it is designed to work in networks with asymmetric connections.

A separate class of protocols are those which we could call *explicit hybridization*. In these protocols there are two, clearly separable routing models, which are frequently full featured routing protocols on their own. The main challenge of these protocols, naturally, is the appropriate choice between the two protocols as well as their integration in the systems - the basis of this decision is frequently the defining factor of the protocol. For instance, LRHR[129] takes the decision based on link reliability, while Bamis et al. [9] makes the decision based on mobility classes. There is also a wide variety among the protocols combined: Zone routing with ant colony optimization in HOPNET[122], FSR with ZRP in FZRP[133], reactive ants with proactive ants in ANSI [97].

Table 3.3.1 summarizes the protocols reviewed in this section and compares some of their features.

3.4. 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



(a) (b)

Figure 21: LAR routing protocol. The diagrams (a) and (b) present LAR1 and LAR2 schemes [62, 63].

location. This location information is then utilized by the routing protocol to determine the routes.

Location-Aided Routing (LAR) [62, 63]: Ko and Vaidya present the LAR protocol which utilizes location information to minimize the search space for route discovery towards the destination node. LAR aims 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 must 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 must be increased to include as many neighboring nodes as possible.

LAR defines two schemes to identify whether a node is within the request zone (see Figure 21).

- *Scheme 1*: The source node simply includes the smallest rectangle containing the current 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*: 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.

Distance Routing Effect Algorithm for Mobility (DREAM) [11]: Basagni et al. propose the DREAM protocol which also uses the node location information from GPS systems for communication.

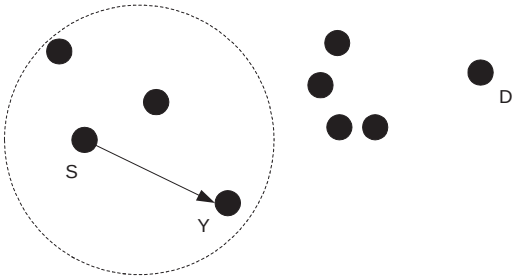


Figure 22: Y is S's closest neighbor in greedy forwarding [59].

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. 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. 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.

Greedy Perimeter Stateless Routing (GPSR) [59]: GPSR, by Karp and Kung, 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 (see Figure 22). 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.

Dynamic route maintenance (DRM) for geographic forwarding [30]: Chou at al. propose a dynamic beaconing scheme to be used in geographic forwarding algorithms in MANETs. In beacon based protocols, each mobile node transmits periodic beacons to its neighbors to update and maintain its routing table. The beacons are generally forwarded at fixed intervals of time. During low mobility, a longer interval would be the best as it would reduce control overhead while providing accurate location information. However, in cases of higher mobility, determining an appropriate beacon interval is rather difficult. In DRM, beacon interval and route information are carried out dynamically. Based on the node's mobility information, its beacon interval is computed while the route management function updates the routing table entries. The DRM algorithm is applied to GPSR forwarding algorithm. The results show that usage of DRM reduced the cost of route maintenance in scenarios with low mobility and improved packet delivery rates where mobility among the nodes is higher.

Improvements to Location-Aided Routing Through Directional Count Restrictions [32]: Colagrosso et al. aims to reduce the control packet overhead by reducing duplicate route formation

packets. The enhancements are proposed to the LAR Box algorithm which is based on count restriction [126] of rebroadcasts. A node after receiving the route request packet waits for an assessment delay (AD) time interval before deciding whether to rebroadcast the packet or simply drop it. This decision is base on the number of duplicate broadcasts received during the AD interval and if this number is greater than a count threshold, then the packet is simply dropped. Using the count threshold value, the number of control packets can be reduced by not rebroadcasting them.

Adaptive Location Aided Mobile Ad Hoc Network Routing (ALARM) [19]: The Adaptive Location Routing (ALARM) algorithm, by Boleng and Camp, uses feedback for adaptation and location information for performance improvements. While using location information has shown to increase efficiency, feedback is suggested as a mobility metric assisting ad hoc network protocols adapt to the current network scenario [18]. Link duration is considered as a suitable mobility metric since constant links for long periods denote low mobility while links experiencing shorter average durations represent high mobility scenario. Essentialy, the link duration feedback agent allows the proposed protocol to become adaptive. The ALARM protocol aims to optimize the existing protocols and devise techniques to combine multiple protocols into a hybrid protocol where more suitable techniques based on the present network conditions can be employed.

ALARM uses the link duration feedback at each of the mobile node to determine the appropriate forwarding scheme. A threshold duration is provided to make further decisions. ALARM forwards the packets on the specified path if the link duration of the source node is larger than the threshold, otherwise it performs a directed broadcast. In such a scheme, even though a broadcast flood is initiated, this flood may be "damped" by nodes closer to the destination if they have stronger links. Hence, the propagation method of the packets is selected based on the mobility metric, the link duration. The important thing is to deliver the packet through regions of "high instability" and it does not matter whether the packet reaches the final destination via flooding or through the original route selected. The *flood horizon* parameter limits the number of hops to flood a packet, keeping the floods within a specific horizon. No route repairs are required in this scheme since the packets can be forwarded over network "hot-spots" by directed flooding resulting in decreased overall end-to-end delay.

A Region-Based Routing Protocol for Wireless Mobile Ad Hoc Networks (REGR) [76]: The REGR protocol, proposed by Liu et al., dynamically creates a pre-routing region between the source and the destination, hence control the flooding of route request packets within this region. The correct selection of the region, which should not be too small, is important for the discovery of the optimal routes.

Once the pre-routing region is selected, all nodes within this region engage in forwarding route request packets. The two main characteristics of this protocol include a *region based route creation* and a *region based route update*. While the route creation is carried out by using existing broadcast schemes, it is followed by creation of a routing region in the neighborhood of this preliminary route. When a route update is needed, this routing region is used to propagate the route update packets. Route creation consists of the following phases:

Destination discovery: A destination location packet (`DLOC`) is first broadcast by the source node. Any existing broadcast algorithm can be employed in this phase. All nodes forwarding the `DLOC` store the path, so when the destination is reached a preliminary route is ready.

Formation of the pre-routing region: Once the preliminary route is selected, the destination broadcasts the Region Definition (`RDEF`) packets which contain the address of the previous hop from the preliminary route and a REGION-WIDTH value which defines the maximum number of hops from the nodes on the route. The nodes receiving the forwarded `RDEF` packets check whether they fall within this range or not. If so, they are included within the region.

In-region route discovery: A short period after the `RDEF` packet is broadcast, the destination again broadcasts a RREQ packet. Those nodes which are marked within the region by the `RDEF` packets re-broadcast the `RREQ` packets. Such a technique helps in limiting the route discovery procedure to a specific region resulting in decreased control overhead.

Since the route selected is currently in use, the route discovery process is not needed. Route update packets are flooded throughout the accepted region to update the route changes.

Location Aided Knowledge Extraction Routing for Mobile Ad Hoc Networks (LAKER) [70]: Li and Mohapatra The LAKER protocol, by Li and Mohapatra, minimizes the network overhead during

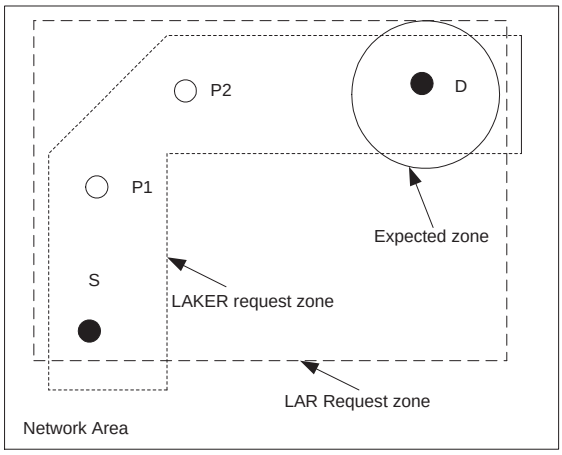


Figure 23: Request zone LAKER vs LAR [70]. It can be seen that the request zone in LAKER is more specific in comparison with LAR; therefore, there is a greater probability of accuracy in determining the exact location of the destination node.

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, the mobility model chosen becomes very important. The restricted random waypoint mobility model [15], which is an extension of [21] is used. LAKER uses an on-demand request-reply mechanism for route discovery (see Figure 23). The control packet format in LAKER contains its forwarding route and guiding route metrics in order to decrease the forwarding area of these route request packets. Knowledge extraction is achieved by keeping track of the number of neighbors of each node till a certain threshold value is reached. The route reply packet forwards the forwarding and guiding routes to the source.

A Location-based Routing Method for Mobile Ad hoc Networks [14]: Blazevic et al. propose *Terminode Routing*, a combination of a location-based routing protocol called Terminode Remote Routing (TRR) and a link state routing called Terminode Local Routing (TLR). TRR is used for nodes located some distance away from the source node, while TLR is used for local nodes. Terminode routing also uses a unique flooding scheme called Restricted Local Flooding (RLF) for flooding control packets during route discovery. *Anchors* are geographical points serve as pointers for source nodes to route the packets.

Naturally, the location independent addressing is used by TLR; on the other hand, TRR uses direct paths, perimeter modes and anchors. A direct path is an approximation of a straight line from the source to the destination while perimeter modes help to turn around obstacles if the packet is "stuck" at a location with no neighbor node closer to the destination than itself.

Perimeter modes often result in the discovery of suboptimal paths since there may be multiple routing loops created. The usage of anchors can avoid this problem. Anchors are referred as imaginary static geographical locations, not the nodes within the network. These locations are written to the packet header and the packet is forwarded accordingly to the intermediate node closest to the anchor. A well chosen set of anchors could decrease the total number of hops in the route. An addtional technique is proposed to allow the source node to check whether anchors are needed or not depending on an estimate of the number of hops on the direct, non-anchored path. Anchors are selected using the Friend Assisted Path Discovery (FAPD) or the Geographical Map-Based Path Discovery (GMPD). While FAPD responders have a stable view of the network, GPMD assumes the availability of the network density maps to the source nodes. Even though GPMD performs better, it requires an extra overhead

of distributing the maps in the network.

Movement-Based Algorithm for Ad Hoc Networks (MORA) [16]: MORA, by Boato and Granelli, takes into account the direction of the movement of the neighboring nodes in addition to forwarding packets based on the location information. The metric 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*: The Distributed MORA 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.

On-demand geographic path routing (OGPR) [46]: Giruka and Singhal propose a geographic path routing protocol which does not depend on a location service to find the position of the destination. OGPR is stateless and uses greedy forwarding, reactive route discovery and source-based routing. It is a hybrid protocol incorporating the effective techniques of other well known routing protocols for MANETs. OGPR constructs *geographic paths* to route packets between a source and a destination node. A geographic path is termed as a *collection of geographic positions from source to destination* which decouple node IDs from the path, meaning that explicit node IDs are not used to construct a path. The addressing scheme is implemented by using a grid-based position encoding strategy. The entire network area is divided into square grids with a unit square grid assigned as an *order-1* grid. Four adjacent order-1 grids form an order-2 grid and so on. With the grid structure set up, a unique binary address is assigned to all order-1 grids. Such a technique helps decouple node IDs from the addressing scheme and enables any node along the geographic path to forward data packets.

Path discovery is carried out on-demand by the source using a flooding path request (`PREQ`) message. This message accumulates the grid addresses of the intermediate nodes till it reaches the destination node. The path reply (`PREP`) containing the geographic path is forwarded back to the source node. If path breaks, the nodes use a path-healing technique. Every packet traces the geographic path followed from source to destination. By greedy forwarding along the geographic path, multiple paths may be discovered and when the destination receives a packet from a newer path, it sends a path update (`PUPD`) message. Control overhead in OGPR is low and paths found are also loop free.

Secure position-based routing protocol [111]: Song et al. propose a secure geographic forwarding (SGF) algorithm which provides source authentication, neighbor authentication, and message integrity. It is combined with a secure grid location service (SGLS) to enable any receiver to verify the correctness of the location messages. SGF uses both greedy and directional flooding with unicast messages being encrypted with pair-wise shared keys between source and destination. The SGLS provides additional security mechanisms to the original GLS, incoporates secure location querying and secure `HELLO` message exchanges. The additional security features prevents message tampering, dropping, falsified injection, and replay attacks.

Sociological orbit aware location approximation and routing (SOLAR) [45]: Ghosh et al. first propose a macro-level mobility framework termed ORBIT. It is a deterministic orbital movement pattern

of mobile users along specific places called *hubs*. The movement pattern is based on the fact that most mobile nodes are not truly random in their movements but actually move around in an *orbit* from hub to hub. Each hub may be a rectangle and movement may take place either inside a hub or in between hubs. Example orbital models discussed are random orbit, uniform orbit, restricted orbit, and overlaid orbit.

The SOLAR protocol uses the ORBIT framework and the spatial/temporal locality of the nodes around these hubs. It uses the concept of *peer collaboration* among acquaintances (nodes whose hub lists are cached by the node). Each node only needs to know the terrain in terms of hub locations and its own location. Periodic `HELLO` messages are forwarded to neighbor nodes for both neighbor discovery and sharing of hub lists. When a source needs to send data to an acquaintance whose hub list is at the source, the packet is forwarded to the center of the hub. Packet forwarding is achieved in a greedy fashion. If nothing is known about the destination's hub list, a new query is forwarded to the hubs in the acquaintance's hub list. This is called a *logical hop*. The query is forwarded along logical hops till either the destination's hub list is obtained or the number of logical hops exceeds a given threshold. If the query reaches the destination, it responds with its own hub list and its current hub where the packet is forwarded.

Load balanced local shortest path (LBLSP) routing [25]: Carlsson and Eager propose a distributed routing algorithm which uses both local shortest path (LSP) and weighted distance gain (WDG) to finalize the forwarding node. The two non-Euclidian distance metrics provide load balanced routing around obstacles and hotspots. Static nodes with lifetimes longer than the time required to route around an obstacle are considered.

The goal is to find the non-Euclidian metrics which are distributed, stateless and conform to Euclidian distances. To achieve this, the network traffic is routed from a source to its destination further from an obstacle on the path instead of routing along the obstacle boundary. In the LSP metric, only a single such obstacle is considered even though there may be multiple obstacles in the network. The source and destination nodes are first determined whether they are in line of sight of each other. If so, the LSP is the Euclidian distance, otherwise, it is calculated as a sum of three distances. Each distance is calculated as the shortest path around an obstacle. Load balancing is achieved by pushing traffic away from the obstacle boundary and avoid creating local hotspots. The WDG metric is used to provide various levels of importance of either using a straight line route or an obstacle avoidance route. By assigning different magnitudes of weight to either of these routes, the WDG metric combines distance gains instead of absolute distances between source and destination.

The LBLSP routing algorithm combines both the LSP and WDG. A packet initially starts in greedy mode and may change its mode to recovery mode till it reaches its destination. In greedy mode, the forwarding node calculates LSP values for itself and its neighbors. For nodes with lower LSP values, their respective WDG values are computed and the one with the largest WDG value is chosen as the next hop. If there are not any nodes with lower LSP values, then the packet mode is changed to recovery mode. In this mode, perimeter routing is carried out till a node with a lower LSP value is found.

Geographic landmark routing (GLR) [84]: The GLR algorithm, by Na and Kim, solves the *blind detouring problem* and the *triangular routing problem* in MANETs. The blind detouring problem occurs when a packet arrives at a dead-end when the next node is blindly selected (see Figure 24). This could result in a longer detour path even when there are actually shorter paths available. GLR solves this by discovering two paths bypassing the void area. One of the paths is from source to destination while the other path is in the reverse direction. These paths are compared and the shorter path is selected.

The triangular routing problem occurs due to path refraction at dead-end nodes. GLR solves this problem by using the concept of *landmark* nodes which are special intermediate nodes identified during path discovery. This landmark node reclaims greedy forwarding after escaping from a dead-end. GLR computes straight sub-paths between the landmarks. GLR has two sub-layers: basic and optimization. GLR uses a regular geographic routing algorithm such as GPSR to forward packets to target nodes in its basic sub-layer. In the optimization sub-layer, information about forward and backward paths such as hop counts and landmark nodes is collected. Then, it selects the shorter detour path and utilizes a loose source routing with the landmarks.

Maximum expectation within transmission range (MER) [64]: Kwon and Shroff propose a packet forwarding algorithm for location aware networks. In most cases, location estimates have significant error

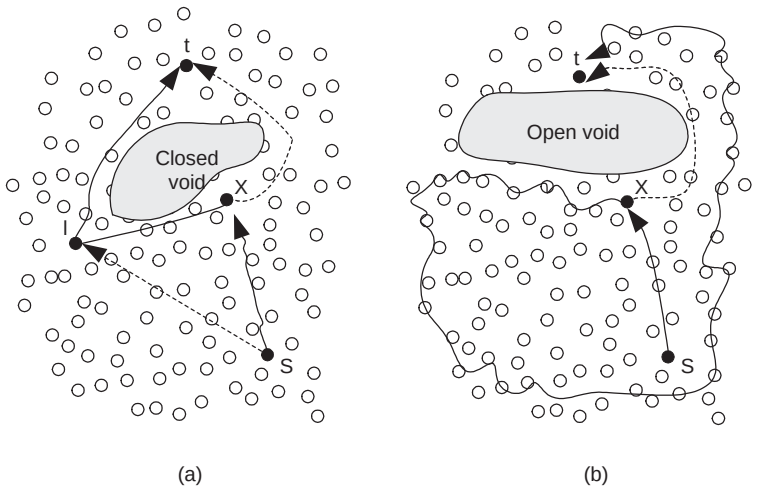


Figure 24: The blind detouring and triangular routing problems, (a) closed void, (b) open void [84].

rates which may be overlooked in most location based routing protocols. These location errors could induce either *transmission failures* or *backward progress* in greedy mode. The former occurs when the selected node is out of transmission range while the latter takes place when the next hop node is actually farther than the destination. This leads to looping within the network.

The proposed MER algorithm utilizes an error information field in its messages which is used by the objective function. The greedy routing scheme (GRS) is improved by the authors where they mitigate the impact of the location errors. Using the error statistic, the objective function calculates a maximum expectation for each node in which the nodes with the highest expectation are chosen as the next node for packet forwarding.

Implementation framework for trajectory based routing (TBR) [137]: Yuksel et al. study various implementation issues of TBR in this work. A proposed method encodes trajectories into packets at the source node before sending them to the destination. Bezier curves are utilized as possible path trajectories to efficiently forward the packets. These curves provide flexibility in the greedy forwarding of TBR with the possibility of multiple types of curves. With a given node neighborhood and a trajectory for packet forwarding, TBR specifies the following objectives:

- Obey the trajectory: The packet stays within the given trajectory and does not deviate at all.
- Reach the destination: If application is more sensitive to packet loss, then the packets should be forwarded directly to the destination node instead of following the given trajectory.
- Reach quickly: The packets are forwarded such that they reach the destination with the minimal delay.

Forwarding nodes on the curve may be selected at random or the nodes closest to the curve (CTC). For flooding applications, the least advancement on curve (LAC) ensures that the maximum nodes receive the packet while a combining CTC and LAC may be another possibility for packet forwarding. To reduce delivery delay, the packet may be forwarded to the farthest node on the curve with most advancement on curve (MAC) algorithm and if the trajectory must be strictly obeyed then the lowest deviation from curve (LDC) can be implemented. In this case, the route deviates from the trajectory as little as possible.

3.4.1. Comparison

The research on location-aided routing has evolved from the design of the generic routing frameworks to their optimization and the solution of specific problems posed by certain ad hoc network configurations.

The earliest proposed protocols were designed to improve on the generic ad hoc routing protocols by taking advantage of the knowledge of the physical location of the nodes. Most of these protocols can

be classified as source originated, reactive routing: LAR[62, 63], DREAM[11], GPSR[59], but they also require the distribution of the location information (and in some cases, movement information as well).

In this respect, these protocols are proactive in distributing location information. However, location information involves less data than full topological connectivity and has a more predictable and continuous change pattern. Nevertheless, the distribution of the location information is a significant overhead and the DRM[30] and improvement to LAR[62, 63] approaches are directed towards reducing it.

Another group of protocols are based on the general approach that location based routing is used to get to the general vicinity ("region") of the destination, where the routing will be shifted to a different approach. Examples of this group are REGR[76], LAKER[70], and Terminode routing [14].

Another group involves algorithms which resemble source routing from the wired IP networks, but with the explicit list of IP addresses to be traversed replaced with the geographical path that needs to be followed or approximated by the packet. Examples of this approach are on demand geographic path routing (OGRP)[46] and trajectory based routing (TBR)[137].

Finally, one of the most recent focus on location aided routing was an increasing attention paid to the actual distribution of the nodes in the physical space. The particularities of the network occasionally include aspects which are helpful for routing - for instance the SOLAR[45] algorithm which exploits the recurrent movement pattern of the nodes. In many cases, however, the physical space of the ad hoc network contains obstacles or simply areas where no forwarding nodes are present. Following a greedy location-based scheme would lead the packets into a dead end in this situation. Protocols such as LBLSP[25] and GLR[84] have been designed to address exactly these challenges, while MER[64] is designed to reduce the impact of location estimation errors.

Figure 4 summarizes the protocols surveyed in this section.

3.5. Multipath protocols

Multipath routing protocols create multiple routes from source to destination instead of the conventional single route discovered by other protocols. 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 networks congestion which may arise due to bursty traffic within the network. 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) [118]: Valera et al. propose the CHAMP protocol which 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 starts with a zero from the source and is increased by one with every retransmission.

Ad hoc On-demand Multipath Distance Vector routing (AOMDV) [79]: Marina and Das present the AOMDV protocol, which 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 25). 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

Table 4: Geographical routing protocols comparison

Protocol	**Forwarding strategy**	**Route metric**	**Loop-free**	**Scalability**	**Robustness**	**CO**
LAR	Directional flooding	Hop count	No	No	No	Medium
DREAM	Flooding	Hop count	No	No	No	Low
GPSR	Greedy	SP	Yes	Yes	No	High
Colagrosso et al	Directional flooding	Hop count	No	No	No	Low
ALARM	Directional flooding	Hops and mobility	Yes	Yes	No	Medium
REGR	Directional flooding	SP	Yes	No	No	Low
LAKER	Directional flooding	Hop count	No	No	No	Low
Blazevic et al.	Multipath flooding	Hop count	Yes	Yes	Yes	High
MORA	Greedy	Weighted hop count	No	No	No	High
OGPR	Source routing	SP	Yes	Yes	Yes	Low
Song et al.	Greedy and directional forwarding	Hop count	No	No	No	Medium
SOLAR	Greedy geographic forwarding	SP	No	No	No	Medium
LBLSP	Greedy geographic forwarding	LSP and WDG	Yes	Yes	Yes	Low
GLR	Source routing	SP	Yes	Yes	No	Low
MER	Greedy geographic forwarding	Max. expectation	No	Yes	Yes	Low

Route metric: SP = Shortest path; LSP = Local shortest path; WDG = Weighted distance gain

CO = Communication Overhead

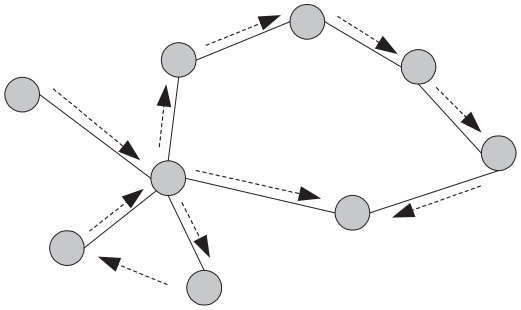


Figure 25: Example of a potential routing loop scenario with multiple path computation [79].

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 [79]: *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 whether it provides a node-disjoint path to the source.

Split Multipath Routing (SMR) [68]: The SMR protocol, by Lee et al., establishes and uses multiple routes of maximally disjoint paths from source to destination. Multiple routes are discovered on-demand and the route with the shortest delay is selected.

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 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) [134]: Yao et al. present an initial theoretical analysis showing that non-disjoint multipath routing has a higher route reliability than the conventional disjoint multipath routing. This study has 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. While k can be set to any value, the control overhead also increases accordingly.

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 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.

Adaptive QoS routing framework through multiple paths [36]: Das et al. observe that when communication takes place between source-destination pairs, an interruption may occur in case of a link breakage due to various reasons. This interruption leads to degraded QoS as the user has to wait till another route is discovered. To solve this problem, an adaptive framework for computing multiple paths in both temporal and spatial domains is proposed. The data transmission in the spatial domain balances the traffic load in the network, while the temporal domain ensures continuity of data transfer.

Two different aspects are considered within this framework: (i) preemptive route rediscoveries are performed before route errors occur as data packets are forwarded from one node to the other. Multiple paths are computed in the temporal domain and in the event of a route error, one of the alternate paths is selected; (ii) multiple paths are also computed in the spatial domain to ensure forwarding of data at any instant of time. These packets are transferred sequentially in blocks over the multiple paths computed, helping to reduce further congestion and link delay.

The concepts of *link stability* and *path stability* are also proposed based on the employed route discovery and route maintenance schemes. Path stability metric depending on the route lifetime should be valid as long as the data is forwarded from source to destination. If the path is not stable enough, then a route maintenance scheme is required to either repair the path or suggest an alternative path. Once the alternative set of paths are decided, the total volume of data is divided into blocks and forwarded along all the different routes to the destination.

Truthful multipath routing protocol (TMRP) [124]: Wang et al. present a multipath routing protocol which can be used in networks with non-cooperative nodes, also termed as *selfish* nodes. These nodes are characterized by the fact that they agree to forward packets only for a return/payment. The cost of forwarding a packet is measured in terms of resources available at each node. A node using the forwarding services of another node would like to minimize its own cost of forwarding while the selfish node may not advertise its true cost. The authors hence design a *truthful* mechanism to ensure that a node's payoff is maximized only when it reveals its true cost.

A generic truthful multipath routing protocol (GTMR) is presented which can be used to make any *table-driven multipath routing protocol a truthful one* without additional control overhead. It only requires the multipath routing protocol to be loop-free and table-driven. The GTMR builds a coordination between the neighbor table and the routing table of the multipath routing protocol and is followed by an auction based approach to select the next hop for packet forwarding. The second-price sealed bid auction, the *Vickrey auction*, is used by a forwarding node when it receives a packet to forward. All one-hop neighbors of the node are qualified bidders who advertise their bids in their `HELLO` messages. Based on the lowest cost, the winner is selected and the packet is forwarded.

The authors implement the GTMR over the AOMDV multipath routing protocol and call it the TMRP. The TMRP has two variations depending on whether the node changes its packet forwarding cost over time or not. In the case of varying the cost, the node varies the cost proportionally to the number of packets forwarded until the cost reaches an upper bound. At this moment, the cost is reduced to its lower bound. Nodes use the routing table and the packet forwarding auction approach suggested in GTMR to decide the next hop where to forward the packet.

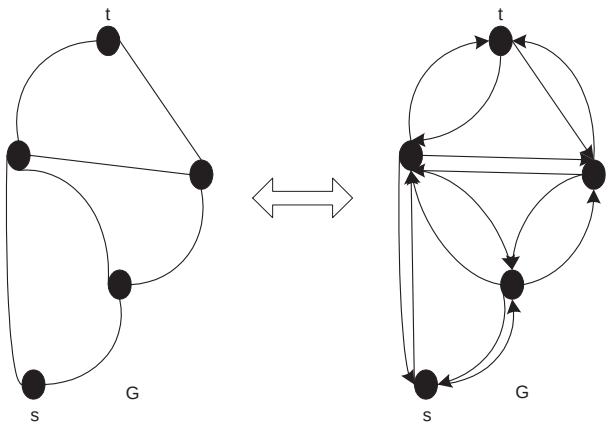


Figure 26: Equivalent graphic representation of an ad hoc network [74].

On-demand discovery of node-disjoint paths [74]: Liu et al. design an algorithm to find node-disjoint paths in a multipath routing protocol for MANETs. The aim is to guarantee discovery of node-disjoint paths. A theoretical framework proves the equivalence between multipath discovery and network flow assignment. A typical flow assigment algorithm (the Ford-Fulkerson method) is considered and with the aid of reverse mapping, the augmenting path in the flow network is mapped to an *auxillary* path in the ad hoc network (see Figure 26).

Multiple node-disjoint paths (MNDP) is proposed, an incremental algorithm where an auxillary path is computed and merged with the existing path set between a source-destination pair. In addition, the auxilary path discovery protocol (APDP) is introduced and integrated with the MNDP algorithm for discovery and maintenance of multiple paths. The APDP is a derivation of DSR and its route discovery is carried out in three phases.

- Phase 1: Initial path from source (s) to destination (t) is discovered using DSR and is termed the *reference* path.
- Phase 2: Auxilary path is discovered based on the reference path.
- Phase 3: Reference and auxilary paths are merged to give two node-disjoint paths.

To compute `k` node-disjoint paths, the phases 2 and 3 must be executed `k` times. Route maintenance is also achieved by executing phases 2 and 3 with an existing valid path as the reference path.

Scalable multipath on-demand routing (SMORT) [102]: Reddy and Raghavan propose a multipath routing protocol with the aim of minimizing route recovery overhead. A primary path from source to destination is chosen and multiple intermediate nodes are selected on the primary path to provide multiple paths. The primary path is most often the shortest path. SMORT has no requirements on discovering *node-disjoint* paths and finds multiple *fail-safe* paths. A fail-safe path is auxillary to the primary path and bypasses at least one intermediate node on the primary path. Multiple fail-safe paths differ from the node-disjoint and link-disjoint paths by the fact that they may have common nodes and links. In fail-safe paths, a new path is selected right away, keeping the control overhead and delay minimal. Route discovery in SMORT is similar to AODV with the only difference being each node is allowed to accept multiple copies of route request packets. To avoid end-to-end routing loops, SMORT does not allow extra route reply packets from the intermediate nodes to be relayed back to the source node.

Secure multipath routing (SecMR) [80]: Mavropodi et al. propose a multipath routing protocol with various security enhancements to guard against collaborating malicious nodes. The SecMR is designed to protect against denial of service (DoS) attacks from malicious nodes. Multiple paths may be affected by several vulnerabilities such as man-in-the-middle attacks, lack of authentication or the racing phenomenon. The SecMR guards against these vulnerabilities and discovers existing non-cyclic,

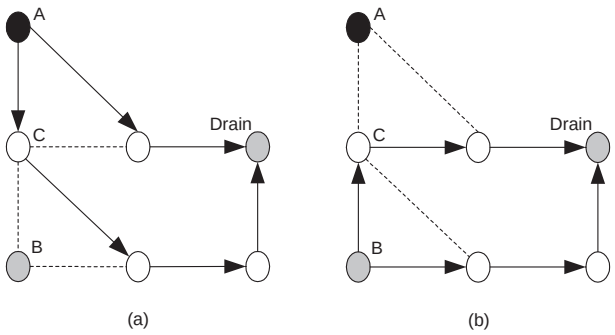


Figure 27: Two node disjoint path from nodes A and B to the drain, computed independently by each node [99].

node-disjoint paths between a source and a destination. In the path selection process, the number of hops is required not to exceed a specific maximum.

The SecMR is divided into two main phases: neighborhood authentication and route discovery and maintenance. The former implements the asynchronous mutual authentication of nodes in the neighborhood using a pair of public secret keys. Each secret key also must be certified through a certifying authority (CA). Signed messages are broadcast to immediate neighbors by each node at periodic time intervals containing the current time and its unique identifier. In the route discovery, the maximum hop count is determined before the random key selection. The encrypted key is calculated before the message construction and then the message is broadcasted. Every route request query comprises of a different independent thread received at the destination. The destination node decrypts the message with the secret key to check the validity of the received message. The communication is initiated through node-disjoint routing paths.

Disjoint multipath routing using colored trees [99]: Ramasubramanian et al. develop a multipath routing protocol using a pair of colored trees. The proposed red and blue trees are either link disjoint or node disjoint (see Figure 27). Every node in the network maintains two preferred neighbors for every destination. These neighbors compose the respective red and blue colored trees. The distributed algorithm uses only local information to compute multiple disjoint paths to a destination node. The *drain* node, an intermediate or final destination is the node in which the data is forwarded along the trees. Any packet transmitted from the source is marked with either red or blue colors. The intermediate node receiving this packet forwards it to the preferred neighbor based on the color of the packet. The colored trees are constructed using the depth first search (DFS) technique. A node in the network could be in one of the following states at any point of time: unvisted, visited, cycle, token and finish. Nodes in the cycle state belong to both trees and those in the token state initiate path searches. Once a node receives a token message, it initiates a path search by sending out *search* messages. These messages are forwarded sequentially to every node in the neighbor list until a *success* message is received from the neighbor. A node in the cycle state sends this success message. Token messages are then sent to select nodes from which the success message was received and the neighbor list traversed in the reverse direction. Every node finishing the operation sends a *return* message denoting the end of the operation. After all neighbors send the return messages, a final return message is sent to the parent node initiating the path search with a token message. This procedure results in the creation of a pair of colored trees from the originating source to the drain node.

Reliable and efficient forwarding (REEF) [33]: Conti et al. propose an efficient forwarding, based on reputation and reliability, scheme to be used in conjunction with an existing multipath routing protocol in MANETs. The mechanism is trustworthy and builds the reputation of each node with a set of forwarding policies while avoiding unreliable routes and balance network utilization simultaneously. REEF uses only the node's internal knowledge. The basic principle followed states that *cooperation problems should not be solved by using cooperation.* This means that each node trusts only itself. REEF uses an existing multipath routing protocol to obtain information such as the next hop and the number of hops towards the destination. Packets are forwarded on a route using either a probabilistic scheme or simply the best/most reliable route. The protocol basically

forwards the packet on the route with the highest success probability. The novelty of this protocol lies in the fact that the data forwarding is accomplished in a lightweight manner with no additional overhead.

Multipath security-aware QoS routing (MuSeQoR) [103]: Reddy et al. propose a multipath routing protocol with QoS guarantees ensuring secure and reliable communication even in the case of multiple path failures. Multiple paths are selected based on the current state of the network and the number of possible paths present between source and destination nodes. MuSeQoR considers two types of channel models: *erasure* channels and *corruption* channels. The former deals with channels susceptible to path loss due to link failures while the latter considers channels which may be corrupted by malicious nodes. The protocol is a modification of DSR with no global topology information maintenance at the nodes. Sessions are divided into time frames during which it is assumed that the network state remains constant. Based on the network state during a session, paths are computed. During route discovery, route request (`REQUEST`) packets are sent by the source. These packets contain a sequence number, the path traversed and the reliability of the path traversed till now, the source and destination IDs, the path bandwidth and the required bandwidth for this session, the path failure metric, eavesdropping ratio for this session. Intermediate nodes keep checking the packet route to make sure no loops are created. `REQUEST` packets are forwarded a predefined number of times after link and bandwidth availability on the path is recalculated. Once the destination receives multiple `REQUEST` packets, each path is sorted by their failure probabilities. Using the position and velocity information of the last node that forwarded the packet, link availability is estimated. Once a set of paths is obtained, each path receives a `RESERVE` packet. The source begins data transmission along the path in which the `RESERVE` is received. However, the destination sends a `RESERVE` packet only if a set of node-disjoint paths are found.

3.5.1. Comparison

Multipath protocols find multiple paths between the source and the destination and use these paths to improve on specific performance metrics. The papers discussed in this section cover a very wide range of approaches. There are, however, several discernible trends.

A certain group of papers are striving to add multipath extensions to existing protocols. Thus, AOMDV[79], TMRP[124] and SMORT[102] build on AODV, while the work on on demand discovery of node disjoint paths [74] and MuSeQoR[103] build on DSR. Note that both are source originated protocols - in general, most of the approaches in this class had chosen a source originated approach. Several approaches are also using source routing (such as split multipath routing (SMR)[68]), giving the source complete freedom over the path to be followed by the individual packets.

The paths in the multipath protocols must meet individually performance requirements, similar to single path protocols. In addition, the routes have specific constraints on their relationship: they must be either link-disjoint or node-disjoint, although other criteria are also possible, such as described in NTBR[134]. This is a much more complex optimization problem than, for instance, finding a single shortest path. Among the papers surveyed, several are using sophisticated mathematical models for optimizing multiple paths: e.g. Lagrangian relaxation and subgradient heuristics (Adaptive QoS routing [36]), network flow theory (on demand discovery of disjoint paths [74]) and colored tree graphs ([99]).

We get a different cross-section of the field if we consider the ultimate goal of building multiple paths. Surprisingly, relatively few paper consider load balancing as one of the primary goals (for instance, SMR[68] and adaptive QoS routing [36]). For most papers surveyed, the main objective of the multipath approach is fault tolerance and/or quick recovery from route failures: CHAMP[118], NTBR[134], adaptive QoS routing [36] and SMORT[102].

A somewhat different approach is taken by papers which use multipath routing to improve the overall security of the transmission and to defend against malicious nodes (SecMR[80] and MuSeQoR[103]) or to improve the efficiency of the forwarding in networks with unreliable nodes (REEF[33]).

3.6. Hierarchical protocols

While traditional internet routing is natively hierarchical, in the first approximation ad hoc networks route over a flat collection of nodes. As the networks grow in size, these approaches lead to increased routing table sizes and control packet overhead. Hierarchical ad hoc routing protocols build a hierarchy of nodes, typically through clustering techniques. Nodes at the higher levels of the hierarchy provide special services, improving the scalability and the efficiency of routing.

Table 5: Multipath routing protocols comparison

Protocol	Proactive/Reactive	Loops	Route metric	Route cache
CHAMP	Reactive	Yes	Shortest path	Yes
AOMDV	Reactive	No	Advertised hopcount	No
SMR	Reactive	No	Least delay	No, table at source
NTBR	Reactive	Yes	Link active	Yes
Das et al.	Reactive	No	Least delay	Yes
TMRP	Reactive	No	Auction winner	No
Liu et al.	Reactive	No	Shortest path	Yes
SMORT	Reactive	No	Shortest path	Yes
Ramasubramanian et al.	Proactive	No	Preferred neighbor	RT
MuSeQoR	Reactive	No	Shortest Path	Yes

Hierarchical State Routing (HSR) [53]: Iwata et al. introduce a class of protocols based on multilevel clustering. The goal is to replace the flooding of the control information with a local collection of this information in the clusterhead, followed by the propagation of this information to the other clusterheads.

First, nodes form level 0 clusters based on physical proximity and elect a clusterhead. Clusterheads connect to each other using *virtual links*. Multiple clusterheads can assemble themselves into higher level clusters. When a node changes its position, link state information is exchanged between clusterheads using the virtual links. The clusterhead collects link state information about the nodes in its cluster and propagates it to other clusterheads through *gateway nodes*.

Routing in HSR happens using a hierarchical addressing scheme, with the clusterhead acting as routers. When a node wants to send a packet, it sends it first to the local clusterhead. The clusterhead looks up the destination and sends the packet to its nearest gateway node. The gateway node then propagates the packet to the nearest gateway node at the next level of the hierarchy. The process continues until the packet reaches the gateway node of the destination cluster. The final gateway node routes the packet to the clusterhead of the destination cluster which then forwards the packet to the destination node.

Core-Extraction Distributed Ad Hoc Routing (CEDAR) [110]: The CEDAR protocol, proposed by Sivakumar et al. is allows the consideration of QoS requirements in an ad hoc setting. The protocol selects a subset of nodes called the *core* of the network. Control messages will only be broadcasted among the nodes of the core, which can use any existing ad hoc routing mechanism for communication. The core is positioned as a "self-organizing routing infrastructure" which performs route availability computations through waves of messages with dynamically limited propagation speed. The availability of increased bandwidth is transmitted by slow propagating *increase waves*, while information about decreased bandwidth is transmitted by fast propagating *decrease waves*.

Routing in the CEDAR architecture happens as follows. The source node sends a route request packet containing the source, destination, and the requested bandwidth to its *dominator*, the local core node. The dominator then computes and establishes a QoS route if feasible. The dominator nodes in each cluster maintain local state information and communicate with each other using virtual links. Route computation is carried out only on the core path.

CEDAR aims more at robustness than optimality in computing the routes. Each core node only knows about the neighboring core node and has no global knowledge about the core subgraph. This simplifies the maintenance of the core network which can be necessary due to topology changes induced by the mobility or failure of nodes. Core paths are established on-demand for connection requests and route computation is carried out only when a specific request for a route is received.

Eriksson et al.'s Dynamic Addressing Approach [39]: Eriksson et al. propose a dynamic addressing scheme to improve scalability in ad hoc networks. The approach is based on adding a location based dynamic address to the node, in addition to its permanent identifier. A distributed lookup table is used to map the permanent identifier to the dynamic address (see Figure 29). The main challenge is the assignment and maintenance of dynamic addresses, which is done through a hierarchical address tree. The approach had been shown to successfully scale to networks of several thousands of

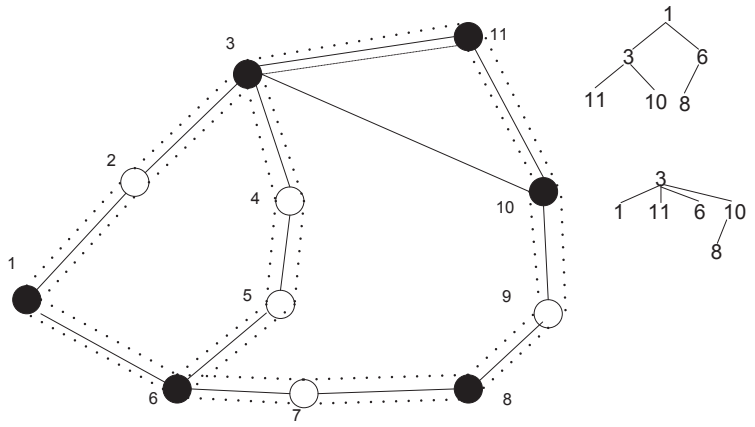


Figure 28: Example of a core broadcast. Nodes in black are the core nodes. Solid lines denote links in the ad hoc network. Dotted pipes denote virtual links in the core graph [110].

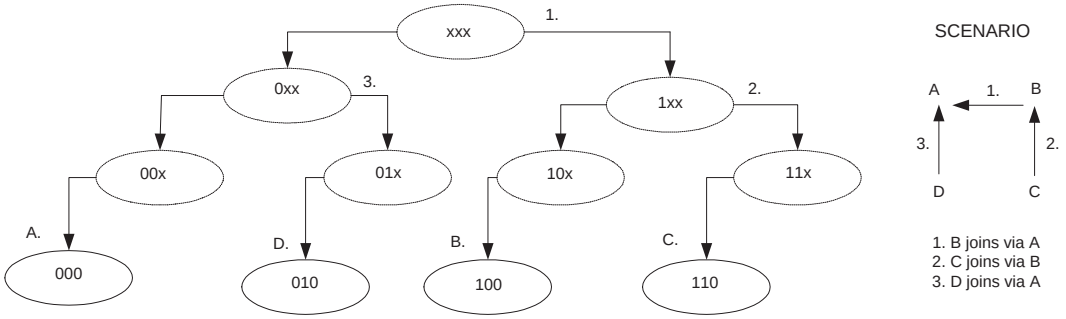


Figure 29: Address tree for a small network topology. The numbers 1-3 show the order in which the nodes were added to the network [39].

nodes.

Hierarchical Landmark routing (H-LANMAR) [131]: An extension to the LANMAR scheme [91] is proposed by Xu et al. The LANMAR protocol is logically hierarchical and uses *landmarks* which summarize routing information to remote nodes. H-LANMAR improves the scalability of LANMAR through the use of a *backbone network*. In the first step, nodes are grouped into dynamic multi-hop clusters, with a clusterhead called the *backbone node* (BN). Backbone nodes are then connected using higher-level links and the process is continued recursively to create a multilevel hierarchy.

In the original LANMAR protocol the packet is forwarded to the nearest landmark. In contrast, H-LANMAR routes the packet to the nearest BN which then forwards it along the backbone network to the other BNs until the packet reaches the BN of the destination node. This BN then sends the packet either directly to the remote location or to its nearest landmark. This process helps in reducing the number of hops to the destination and the overall packet delay. It is assumed that each node is simultaneously running the LANMAR scheme locally in case of a failure in the backbone.

3.6.1. Comparison

Hierarchical routing protocols, in general are proposed as a scalability approach. In contrast to wired IP routing, the hierarchy has nothing to do with the hierarchical nature of the address, rather the hierarchy reflects a geographical clustering of the nodes.

Eriksson et al. [39] is an unusual case in the sense that it introduces a second, dynamic address for the nodes, based on their geographical location.

Another interesting observation is that all the protocols we have reviewed create the hierarchy dynamically and in a distributed manner. It is certainly possible to create a hierarchy in a centralized way

Table 6: Hierarchical routing protocols comparison

Protocol	Routing tables	Update frequency	Hello Message	Critical node
HSR	Yes	Periodic	Yes	Yes
CEDAR	Yes	On demand	No	Yes
Eriksson et al.	Yes	Periodic	No	No
H-LANMAR	Yes	Periodic	No	Yes

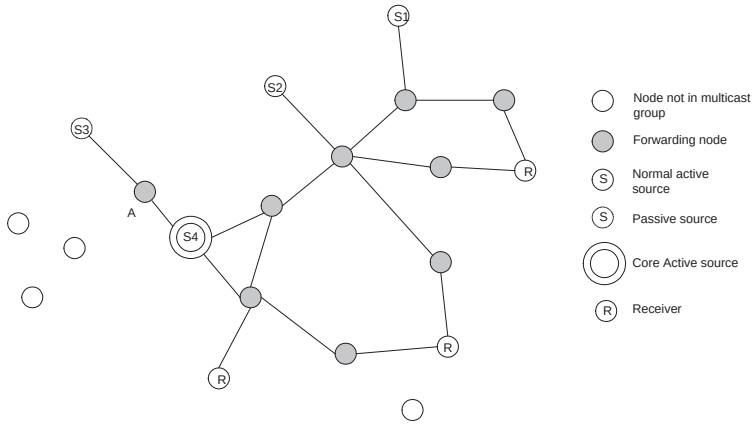


Figure 30: Mesh topology in DCMP [35].

- yet none of the approaches have chosen to do so.

Another way to think about these protocols is whether the protocol have been a hierarchical adaptation of a previously existing protocol (H-LANMAR[131]) or have been designed from scratch to be hierarchical (all the other reviewed protocols).

Another aspect we need to consider is whether QoS considerations such as the current load affect the hierarchical routing or not. From the surveyed protocols only CEDAR[110] considers QoS considerations.

A number of other comparison criteria has been summarized in Table 3.6.1.

3.7. Multicast protocols

Multicasting is the simultaneous transmission of data from one sender to multiple receivers. Several widely used applications require multicasting at least at the logical level. Examples include audio-video teleconferencing, real-time video streaming and the maintenance of distributed databases. In many cases it is advantageous to implement multicasting at the level of the routing algorithm (other approaches would be one-to-all unicast or the implementation of multicasting at the application layer). In the following we review several representative examples of multicast ad hoc routing protocols.

Dynamic Core based Multicast Routing (DCMP) [35]: Das et al. introduces DCMP, a source-initiated multicast protocol. DCMP has been designed from the ground up as a multicast protocol, without relying on existing unicast protocols.

DCMP classifies the sources into *active*, *core active*, and *passive* as shown in Figure 30. Active sources use the traditional technique of flooding the network with `JoinReq` control packets at regular intervals. Nodes which desire to join the multicast group as a destination, reply with a `JoinReply` packet along the reverse path to the source. Passive nodes do not participate in the creation of the multicast routes themselves. Instead, a subset of the active nodes, the core active nodes form a shared mesh through which the passive sources transmit their data packets. A single core active source can support a maximum of *MaxPassSize* passive sources and the hop distance between these sources is limited by the *MaxHop* parameter.

Simulation results show that the control packet overhead of DCMP is up to 30% percent lower compared to multicast routing protocols not employing a core. This leads to an increase in scalability and multicast efficiency of about 10-15%, with a slight tradeoff (2%) in the packet delivery ratio for

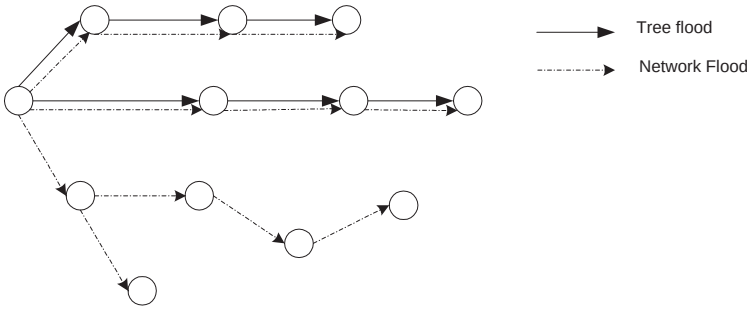


Figure 31: Tree flood vs. network flood [54].

networks with light traffic loads.

Adaptive Demand-Driven Multicast Routing (ADMR) [54]: Jetcheva and Johnson propose ADMR, an on-demand multicast routing algorithm with almost no periodic components within its system. ADMR dynamically maintains the multicast routing state for active groups of nodes. The protocol uses source-based forwarding trees and continuously monitors the traffic pattern of the source. This allows ADMR to detect broken links in a tree as well as to identify the sources which stopped sending data packets. The sender node will transmit "keep-alive" messages to maintain the forwarding tree. When the source wants to terminate the route, it stops sending the "keep-alive" messages, and the tree expires. Similarly, the receiver nodes need to keep alive their respective branches by sending downstream passive ACK messages. If these messages are stopped, ADMR "prunes" the specific branches of the forwarding tree. Multicast data packets are forwarded from the sender to the multicast receivers using MAC layer multicast transmissions along the path of shortest delay.

In a network with highly mobile nodes the cost of maintaining the quickly changing multicast trees is higher than simply flooding the information to every node and letting the node decide whether it wants to retain it. ADMR detects moments when the overall node mobility is very high and automatically starts flooding the network with data packets. After a short period of time, ADMR switches back to the normal mode since within this time node mobility may have reduced. Controlled flooding is highly beneficial and its relation with regular flooding is shown in Figure 31.

The novel features of the protocol are summarized [54]: (i) no periodic floods of control packets or routing table exchanges and it also does not require any core node; (ii) passive acknowledgments are used to automatically adjust trees; (iii) high node mobility can be detected by ADMR and in such a scenario it simply reverts to flooding of data packets since routes become unstable and change frequently; (iv) a limited number of keep-alive packets are forwarded to bursty sources to ensure that their forwarding trees are maintained as often as possible and distinguish between lack of data or complete disconnection.

AMRoute: Ad Hoc Multicast Routing Protocol [130]: Xie et al. propose AMRoute which aims to avoid the high control packet overhead associated with the maintenance of multicast trees in ad hoc networks with highly mobile nodes. It does not support guarantees for minimal bandwidth and packet latency; the main design objectives are robustness and scalability. A conventional unicast routing protocol is used to keep track of the network dynamics. AMRoute is independent of the underlying unicast protocol, which can be chosen according to the specific network requirements.

Thus, AMRoute is only concerned with the dynamics of the multicast groups. The protocol defines user-multicast trees composed of senders and receivers. Packet forwarding is carried out by the members of the groups over the *unicast tunnels* which form the links in the tree. Through this approach, in AMroute nodes which are not members of the multicast trees do not need to support any multicast protocol and hence are not storing any state information.

AMRoute uses a *logical core* to discover new group members, create and maintain the multicast tree. The logical core is not fixed, it changes with the group dynamics.

Energy efficient multicast routing [69]: Li et al. focus on developing an energy efficient multicast routing protocol. By assigning the transmission power of each node as a weight, the network graph is transformed to a new graph with weights between edges. The minimum energy multicast (MEM)

problem is to find the multicast tree whose total energy cost is minimized. The problem now reduces to the directed Steiner tree (DST) problem. This is a known NP-hard problem, and the authors show that it is unlikely that there is an approximation algorithm with a constant performance ratio to the number of nodes in the network. The authors show several heuristic approaches for the problem.

In the node-join-tree (NJT) algorithm a *cover* set containing all non-leaf nodes is built incrementally by selecting the shortest path from the source node to the nodes in the *uncovered* set. The heuristic grows the multicast tree by selecting the nodes with the highest energy efficiency. The authors describe a distributed implementation of the algorithm where nodes have information only about their neighbors.

QoS multicast routing protocol for clustering mobile ad hoc networks (QMRPCAH) [67]: Layuan and Chunlin present a QoS aware multicast protocol for MANETs with clustering. The proposed QMRPCAH protocol allows a node to maintain only local multicast information and a summary of other clusters; it does not require knowledge of the global network. The protocol supports soft QoS without any hard guarantees. There may exist transient periods of time without the required QoS, for instance during periods of congestion, link breakage or packet loss.

In QMRPCAH every node periodically measures the delay on its outgoing links and broadcasts it to the members of its cluster. Thus, every node keeps its intra-cluster routing tables updated. Bridge nodes maintain the inter-cluster routing tables in a similar fashion. When a mobile node enters a new domain, it uses a remote subscription method to subscribe to the new domain and joins the local multicast tree. When a node wants to join a multicast tree a `JoinReq` message is forwarded to the parent bridge node. The bridge appends its own address to the message and forwards it to the top level bridge node if it is not aware of the multicast tree itself. Multicast tree (MT) messsages are forwarded from the top bridge node to the local node.

QMRPCAH uses a receiver-initiated selection flooding algorithm where links violating bandwidth constraints are deleted. Flooded messages also avoid these violating links.

QoS multicast routing using multiple paths/trees [128]: Wu and Jia propose a routing protocol using multiple parallel paths or trees to ensure the bandwidth requirement of a connection. The protocol is distributed and uses standard route discovery and route reply techniques. The QoS requirements include a bandwidth requirement for a route and a delay bound represented by the number of hops from source to destination. Similar to other on demand protocols, `RREQ` packets are broadcast from the source and `RREP` packets are returned by the destination. Since the destination may receive multiple `RREQ` packets, it has multiple possible routes to the source. It runs a paths/tree selection algorithm to construct the multicast trees. Three different tree selection algorithms are proposed:

- Shortest path tree based multiple-paths (SPTM): It computes the shortest path in terms of hops between source and destination. SPTM uses multiple paths from the shortest path tree (SPT), that a branch in the SPT consists of several parallel paths.
- Least cost tree based multiple-paths (LCTM): It uses QoS parameters such as bandwidth and delay to compute a delay-bounded least cost tree. The source node is the start of the tree and the destination node is added if it presents the least network cost.
- Multiple least cost trees (MLCT): The source searches for least cost trees (LCT) until the aggregate bandwidth for all LCTs satisfy the net bandwidth requirement. Due to multiple parallel trees the delay variance could be large, but the highest delay value is maintained within the delay bound.

Genetic algorithms for group multicast [100]: Randaccio and Atzori propose a genetic algorithm (GA) based approach to the problem of finding multicast trees which optimize bandwidth and delay parameters.

The algorithm is initialized by building a population of multicast trees in isolation by combining unicast paths between the source and destination pairs. The unicast paths follow the shortest path in terms of hops, calculated using Dijkstra's algorithm. From this initial population, the GA algorithm generates various (possibly sub-optimal) combinations and selects them for fitness. The fitness function used by the GA is based on the weighted average of transmission delay and network resource utilization.

Fireworks [66]: Law et al. propose a 2-tier multicast/broadcast routing protocol, Fireworks, which adapts itself based on network topology and group density. At appropriate times, it resorts to broadcast

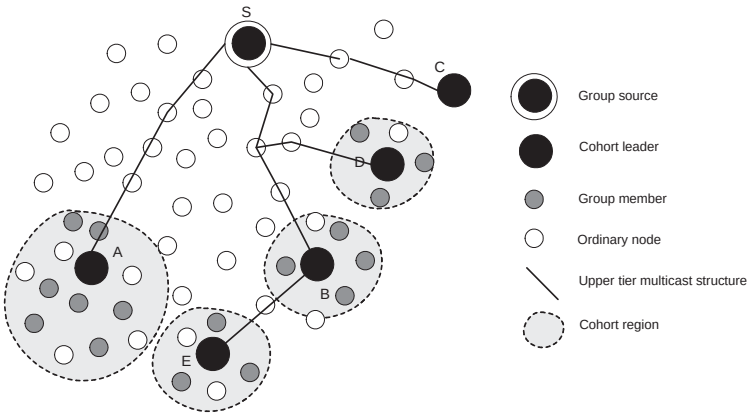


Figure 32: Fireworks 2-tier multicast hierarchy structure [66].

instead of multicast. Sensor nodes are grouped together with local group leaders or *cohort leaders* corresponding to areas of high group member affinity. These cohort leaders establish a sparse multicast tree between themselves and the source node while they broadcast messages within the local group members within their own cohort. The 2-tier hierarchical structure comprises of the *upper tier* formed the source and cohort leaders (see Figure 32). The *lower tier* consists of the members in the cohort. The authors use a new metric termed as *cohesiveness* which maintains the affinity of group members within a node's k-hop radius. Individual group members are discovered using special `ADVERTISE` messages which contain the address, multicast address, hopcount and cohesiveness of the node. The joining node then determines whether it should join as the cohort leader for its k-hop neighborhood. The node becomes a cohort leader if it does not find any cohort leader and broadcasts a `LEADER` message. The upper tier structure is constructed by the source sending `SOURCE-QUERY` messages which are replied to by cohort leaders using `SOURCE-REPLY` messages. Within group members, Fireworks uses broadcasting to forward data while it uses multicasting between cohorts.

Probabilistic predictive multicast algorithm (PPMA) [95]: The Probabilistic Predictive Multicast Algorithm (PPMA) proposed by Pompili and Vittucci improves the robustness and reliability of multicast trees in the event of link and/or node failures. The algorithm defines a new way to quantify the suitability of a link, the *probabilistic link cost* which is comprised of three terms: energy, distance and lifetime. Using this new metric, the multicast trees can be computed in the centralized or distributed manner. In the centralized approach, the algorithm simply substitutes the new metric for the other metrics traditionally used in the centralized Bellman-Form algorithm (such as hop count).

In distributed PPMA, the multicast tree is created based on a private and a public link costs. The private link cost is used by the nodes which are already added to the multicast tree, while the reminder of the nodes in the network, used the public link cost to aggregate paths and form multicast trees.

Hierarchical multicast techniques and scalability [49]: Gui and Mohapatra introduce a framework for hierarchical multicasting in MANET. The proposed approaches include a domain-based and an overlay-driven.

In the domain-based scheme, a large multicast group of nodes is divided into sub-groups. Each sub-group is assigned as a sub-root, chosen based on topological optimality. The sub-root uses its own lower-level multicast protocol to create its tree and deliver packets to nodes within its sub-group. The source nodes of each group and sub-roots form a special sub-group for upper level multicast which is used by the source node to deliver packets to the sub-roots.

In the overlay-driven multicast scheme, a hierarchical overlay multicast protocol is used to construct the virtual multicast tree. In this framework, the upper level multicast tree needs to be logically spanned over all the group members. Once the tree is constructed, each non-leaf node is responsible to forward packets to its children in the tree. This mechanism further improves the data delivery efficiency.

Application layer multicast algorithm (ALMA) [44]: Ge et al. propose an application-layer receiver-driven overlay multicast protocol. As the ALMA protocol operates at the application level, it

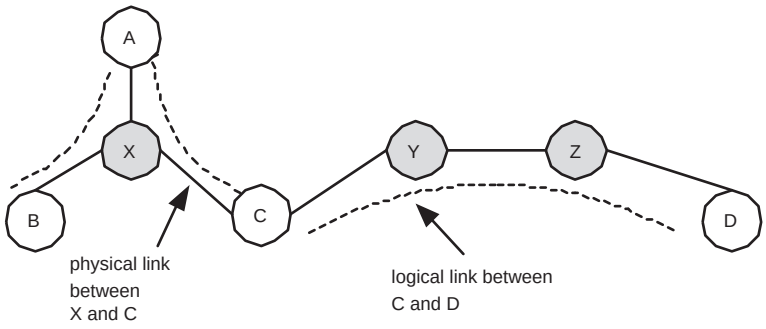


Figure 33: Logical links versus physical links [44].

can be used in conjunction with any routing protocol

ALMA creates a tree of logical links between the group members. If node mobility or congestion makes it necessary, the tree can be dynamically reconfigured. Each edge of the logical multicast tree represents a logical link - a path at the network layer (see Figure 33). The members of the the group can choose to allow zero, one or more children. A new member joins the group by sending *join* messages to multiple existing members. A member willing to take a new child responds and accepts the new node within the tree. If multiple replies are received by the new node, it chooses the member who replies the first. If a member wants to leave a group, it sends an explicit *leave* message to its parents and children nodes. Node failures are treated similarly to unannounced departures from the tree. Thus, each member sends periodic *hello* messages to its parent and receives a return response. If no reply is received in a pre-defined time interval, the nodes assumes the parent has failed and it tries to rejoin the tree. Hello messages also provide a mechanism to measure the quality of a path. Children monitor the quality of the path to their parent and may switch parents when required. With members joining, moving or leaving, the multicast structure is changing and thus the tree requires constant reconfiguration. New parents are searched by nodes based on the average estimated round-trip time (RTT) of a link. New parents should have an RTT less than a pre-specified RTT threshold. Loops can be detected in ALMA but not entirely avoided when two members decide to switch parents at the same time.

QoS aware multicast routing [114]: Sun and Li describe a series of QoS extensions to the MAODV protocol. The approach uses the delay, bandwidth and packet-loss characteristics of MAODV with no additional signaling. It also incorporates multicast routing capability with the existing unicast. A source node sends a QoS route request, `RREQ`, which is forwarded by intermediate nodes until it reaches the destination. The destination sends back the `RREP` packet with a delay time corresponding to a predefined node traversal time (NTT). Intermediate nodes add their own NTTs to the delay value and update their routing tables. Routes with the minimum delay are selected for data transmissions. A similar technique is applied for the bandwidth requirement where source nodes indicate their bandwidth requirements and intermediate nodes compare their available bandwidth before forwarding the packet.

Ad hoc QoS multicasting (AQM) [22]: Bur and Ersoy propose the AQM protocol which tracks QoS availability within the neighborhood of every node based on the requirements and announces it during the session initiation. In order to join a session, the nodes go through a request-reply-reserve procedure that ensures the QoS information is updated and a possible route is selected.

A session is initiated by the initiator (MCN_INIT) node by broadcasting a session initiation packet (`SES_INIT`). This packet consists of the identity number and QoS class of the new session while also setting the bandwidth and hop count rules for the session. Active sessions are maintained in a table (TBL_SESSION) at each node. A membership table (TBL_MEMBER) maintains the status of predecessor nodes. The session information is maintained with periodic session updates keeping track of changes in the QoS conditions and node connectivity. Session update packets (`SES_UPDATE`) refresh the session information periodically while session termination (`SES_TERMINATE`) closes a session. On session termination, nodes clean their tables and free the reserved resources for that session. Neighborhood maintenance is also carried out by broadcasting periodic hello packets (`NBR_HELLO`) which informs neighbors of the node's current bandwidth usage.

Content based multicast (CBM) [138]: Zhou and Singh present a multicast model for a scenario where nodes are interested in obtaining information about specific threats and resources. These threats and resources are a time `t` and distance `d` away from the current location of the node. Nodes generate information about the movement, intensity and location of threats. This information is multicast through the network using a *sensor-push receiver-pull* approach. Here, sensors push the information into the network while receivers pull the relevant information. The network is divided into geographic regions and a sensor detecting a threat broadcasts it into one of these small regions. Individual receivers then pull threat warnings from nodes that lie in the direction of travel. The broadcast message is forwarded in the projected path of the threat to the *leader* nodes of each region. Each leader receiving the message broadcasts it within their groups. Whenever the threat changes direction, a new message is created and rebroadcast. Nodes pull the messages containing the threat information to be warned. Using the `PULL_REQUEST` message, the node requests the leader of the block to send the threat message after a certain time. The leader then sends back all information it has about the threat. These pull requests have a time to live (TTL) field during which the leader sends back threat messages to the node issuing the pull request. This ensures that pull requests are only used infrequently thereby not causing flooding in the network.

Differential destination multicast (DDM) [55]: In the DDM algorithm, proposed by Ji and Corson the source node of a multicast transmission encodes all the destination addresses within each data packet header in an *in-band* fashion. With this approach, no fixed multicast tree is created, the routing will be *soft-state*, similar to state routing algorithms such as DSR. This allows a lower control overhead, as there is no need for extra packets to maintain multicast forwarding state. Control overhead only occurs when there is actual data to send. Nodes along the paths also do not need to maintain alternate backup routes. Once a node receives a data packet, it checks the DDM header to determine which node to forward the packet. This information will be remembered to help in the forwarding of future packets in the same direction. If the destination node changes, the upstream node informs all its immediate neighbors about the difference in the forwarding node.

Robust multicasting in ad hoc networks using trees (ROMANT) [117]: The ROMANT algorithm proposed by Vaishampayan et al. uses a receiver-initiated group joining scheme which does not require any underlying unicast routing protocol or the pre-assignment of cores to groups.

The cores of the groups are determined as follows. When a receiver joins a group, it checks if it has ever received a core announcement for that group. If it did, the node joins the group as a non-core node. Otherwise, the node considers itself to be the core of the group and starts sending core announcement packets with a core ID. If several receivers join the group simultaneously, the one with the highest ID becomes the group core.

The periodically broadcasted core announcements are used to establish a connectivity list at each node, storing information about the core and routes which lead to the core. Router nodes use the core announcements received to determine where to route data packets. Data packets are forwarded to nodes from which the core announcement with the highest ID has been received and the core then forwards the data packet across the tree.

Epidemic-based reliable and adaptive multicast for mobile ad hoc networks (EraMobile) [88]: Ozkasap et al. propose a reliable and adaptive multicast protocol based on bio-inspired epidemic methods. Epidemic methods are *stateless*, thus they are a good match for the rapidly changing, non-deterministic structure of MANETs. The algorithms takes advantage of the broadcast nature of the wireless medium to send gossip messages locally within a multicast group to neighboring nodes.

The traditional approach in gossip based protocols is to select a random node from a predefined list before unicasting the gossip message to the node. In EraMobile, the node gossips with a random subset of one-hop neighbors, constantly changing with node mobility and changes in the local node density. The frequency of gossip messages are also adjusted dynamically.

The periodic gossip messages allow nodes to recover missing data packets and make the protocol robust against delivery failures. The gossip messages also reduce the bandwidth and energy usage and allow a lower control overhead compared to multicast flooding.

3.7.1. Comparison

One of the important comparison terms is whether the multicast happens at the network layer or somewhere else. Most protocols position the implementation of the multicast at the network layer. ALMA[44] implements it at the application layer (more exactly, at what the ISO model would call the session layer - but in our current 4-layer networking hierarchy would be the lowest sublayer of the application layer - with multiple applications being able to be built on top of it).

Most multicast protocols are based on the receiver subscribing to the transmissions of a specific sender. An interesting exception is CBM[138], which performs multicast based on the content rather than the source of the messages.

Almost all the protocols are based on building a multicast tree, although there are some exceptions. CBM[138] does not build a tree due to its radically different distribution model. Differential destination multicast (DDM)[55] performs an on-demand, soft state based multicasting without constructing an explicit tree. Finally, EraMobile[88] replaces the multicast tree with a stateless approach based on epidemic algorithms.

Another question is whether the algorithm is considering the state of the underlying network in the choice of the routing tree. There is an overall group of protocols whose approach is to select a *core* of the network. These nodes will serve as forwarding nodes (for instance, as the non-leaf nodes of the multicast tree). Naturally, the nodes in the core will be nodes with more resources (although other criteria might also be considered - for instance, the fact that the core must extend in all geographic areas of the network). From the protocols reviewed, core based protocols are DCMP[35], AMRoute[130] ("logical core"), Fireworks [66] ("cohort leaders"), and ROMANT[117]. The latter is an example of those protocols where the choice of the core is not based on resources (being simply based on the highest id). Other protocols do not establish a core but consider the available resources of the nodes on a case-by-case basis: energy efficient multicast routing [69] and PPMA[95] where energy is part of the probabilistic link cost.

Most protocols use a distributed implementation, with the only exception being the genetic algorithm based approach [100].

Finally, there is a question whether the protocol considers QoS features such as minimal bandwidth of the multicast. QoS assurance almost always conflicts with resource conservation, as nodes with move advantageous locations or higher bandwidth will tend to become overloaded. From the surveyed protocols, the ones considering QoS are: QMRPCAH[67] (soft Qos), QoS multicast routing using multiple paths/trees [128], QoS aware multicast routing [114], AQM[22].

3.8. Geographical multicast (geocast) protocols

Geographical multicast (geocast) routing is a variant of multicast where the goal is to route the packets coming from a source to destinations located within a specific geographical region. Naturally, for geocast to work, the nodes need to rely on localization techniques (such as GPS). An earlier survey on geocast protocols can be found in [78].

In the following we survey some of the representative geocast algorithms. (To unify the terminology, we shall use the term geocast throughout this survey - different papers use slightly different terminology).

Direction Guided Routing (DGR) [6]: The DGR algorithm, introduced by An and Papavassiliou relies on clusters of nodes, roughly equivalent to the cells in cellular telephony. The clusters are created and the clusterheads elected using dynamic techniques such as Mobile Clustering Algorithm (MBC) [5]. DGR then creates a mesh structure on top of the clusterheads, with the objective to deliver packets within the cluster using reduced overhead. The algorithm for delivering the messages from a source to a specific geocast zone can be described in the following steps:

1. The source formulates the geocast zone, calculates its distance to the center of the zone and sends the geocast messages to its own clusterhead.
2. When the source clusterhead receives a geocast message, it creates and broadcasts a `GeoJOIN REQUEST` to neighboring clusterheads. This is achieved by using the Boundarycast and BoundaryCrosscast operations [51]. When a boundary node receives the `GeoJOIN REQUEST` from its clusterhead, it executes the Boundarycast operation, which is used for routing between clusters.
3. If a non clusterhead receives the `GeoJOIN REQUEST`, it determines whether it is within the geocast zone of the sender by checking the geographical location within the message packet header. If it is within the geocast zone, it simply accepts this message. Otherwise the node makes a decision

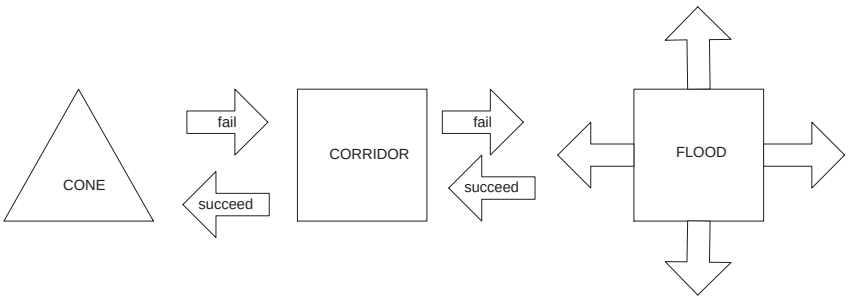


Figure 34: Flow of the FA used to transmit the next JD packet in GAMER [24].

to forward or not based on distance calculations. These reduce the overhead associated with route discovery by limiting the region in which control messages are forwarded.

4. When a clusterhead receives the `GeoJOIN REQUEST`, it stores the clusterhead ID of the earlier clusterhead and performs geocast membership management strategies. If such members are present, it sends `GeoJOIN REPLY` messages to its upstream clusterheads. When these clusterheads receive the messages, they save the clusterhead ID of the node where the message is received and send the message further on upstream. Finally, as the source clusterhead receives the `GeoJOIN REPLY` message, the operation is completed.

Geocast Adaptive Mesh Environment for Routing (GAMER) [24]: Camp and Liu propose the GAMER protocol which creates a dynamic mesh based on the mobility properties of the nodes: highly mobile nodes create a dense mesh, while nodes with a limited mobility create a sparse mesh. The mesh creation approach is based on the approaches by Chiang and Gerla [29] and by Garcia-Luna-Aceves and Madruga [42].

The geocast aspect of GAMER is based on the forwarding zone strategies used in LAR [61] and DREAM [11]. In DREAM, a circle is created which is centered on the last known location of the destination node. The radius of the circle depends on the velocity of the destination node. After defining the forwarding zone, the source node sends the data packet to nodes within one-hop distance of this zone. The nodes then repeat the process by creating their own forwarding zones and forwarding the packets. Therefore, instead of flooding the entire network, the nodes only flood their respective forwarding zones.

GAMER uses a source routing approach based on DSR[57] to route geocast packets around the network. The main problem with source routing is the fact that large amounts of overhead is generated by storing the entire route in the packet itself. Camp and Liu state that GAMER could be modified to store only the local state information instead of source routing information.

Let us now describe the operation of the GAMER protocol.

When the source node in GAMER wants to send geocast packets to transmit, it periodically sends `JOIN-DEMAND` (JD) packets to the geocast region. As all mesh nodes in the target region must join the group, the approach might be more correctly described as geo-broadcasting. The source node chooses one of three possible forwarding approaches (FAs): FLOOD FA, CORRIDOR FA or CONE FA. FLOOD FA floods the network with the JD messages. In CORRIDOR FA, a rectangular forwarding zone is created and the `JD` packets are forwarded through this corridor. Since the area in this case is much smaller than the previous case, a more sparse mesh is created. The CONE FA further restricts the the forwarding zone to an area enclosed by the source node acting as the vertex and the angle created from the tangential sides from this vertex (see Figure 34).

If a mesh node receives a non-duplicate JD packet and it is within the forwarding zone but not within the geocast region, it adds its own address to the source route in the header and forwards it to its neighbors. However, if it is within the geocast region, it responds by sending a `JOIN-TABLE` (JT) packet to its neighbors. This packet contains the reverse source route from the `JD` packet which is then forwarded to the source which can start sending data packets.

A geocasting protocol for mobile ad hoc networks based on GRID (GeoGRID) [73]: Liao et al. base GeoGRID on the unicasting routing protocol GRID [72]. GeoGRID uses location information to define the forwarding zone. The geographic area of the network is divided into logical grids of size d

by d. In each grid area a *gateway* node is elected, whose responsibility is to forward the geocast packets. GeoGRID uses two geocast forwarding methods: flooding based and ticket based.

- Flooding based GeoGRID: In this method, only gateway nodes within the forwarding zone are allowed to broadcast the geocast packets.
- Ticket based GeoGRID: In this technique, only selected gateways are allowed to forward the geocast packets. If the entire region is divided into an m by n region, a total of $m + n$ tickets are first generated. The source then distributes these tickets evenly to neighboring gateway nodes in the forwarding zone which are closer to the geocast region than the source.

The election of the gateway is an important issue in the GeoGRID protocol. The initial gateway of the grid is the node closest to the physical center of the grid. One way in which the gateway of a grid might change is when the gateway node moves out of the grid. Alternatively, a gateway node can silently turn itself to a non-gateway node and relinquish control to the other more suitable node who is then elected as the gateway. Another method of electing gateways could be by using the concept of node weights as in [10].

Geocasting in mobile ad hoc networks (GeoTORA) [63]: Ko and Vaidya propose the GeoTORA protocol which builds upon the unicast TORA [89] routing protocol. TORA uses the distributed "link reversal" algorithm and provides multiple routes to the destination. It also uses the concept of "heights" to determine direction of the links. In GeoTORA, the source node anycasts the geocast messages to the geocast group by using TORA. Once any node in the geocast region receives the geocast packet, it floods the packet into the region. This helps in limiting flooding only within its own region.

3.8.1. Comparison of geographical multicast protocols

Geographical multicast involves multicasting to a collection of nodes which are defined as destinations by their physical location. Many protocols in this class are building upon other protocols: often on location aware protocols such as LAR, DREAM and GRID, but the starting point can also be a non-location aware reactive protocol such as TORA.

Despite their various roots, geocast protocols need to solve a common set of challenges: how to define the geocast area and how to efficiently route the messages to all the nodes in that area. The approaches taken by the protocols are very varied. DGR[6] allows the geocast area to be an arbitrary polygon, but for efficiency purposes it approximates it with a circle, ellipse or a rectangle. GAMER[24] routes towards a rectangular geographic area using a mesh of paths which are either unconstrained, or constrained in a rectangle or a cone. For GeoGRID[73] the destination region is a rectangular subset of the grid. The routing can be either minimally constrained by a rectangle covering both the source and the destination rectangle or through a ticketing approach where the number of tickets issued is controlled by the source. Finally, GeoTORA[63] takes an approach where it first forwards the packet to any node in the geocast region through unicast, followed by a local flooding phase where the packet is flooded throughout the geocast area.

3.9. Power-aware protocols

Power-aware routing makes the routing decisions dependent on considerations of the available energy of the nodes. These considerations can be significantly more complicated than simply finding the route with the lowest energy consumption (in fact, the shortest path is almost always the one with the lowest energy consumption). The protocols from this class take into consideration both the heterogeneity of the energy resources of the nodes, as well as the uneven energy consumption due to the topology of the network and the nature of the data flows. For many of these protocols, the ultimate objective is to maximize the lifetime of a network with nodes with limited and fixed energy resources.

Power aware routing in mobile ad hoc networks [109]: In an influential early paper from 1998, Singh et al. argue that routing protocols for ad hoc networks should possibly consider, in addition to the shortness of the paths, one or more from a collection of other metrics which impact power consumption. The paper proposes the following metrics:

Table 7: Multicast and Geo-Multicast routing protocols comparison

Protocol	**Core/Broadcast**	**Route metric**	**Forwarding strategy**	**Route repository**
DCMP	Core	Newest Route	Source routing	RT
ADMR	Neither	Link breaks	Tree based or flooding	RT
AMRoute	Core	Unicast operation	Shared trees	Depends on unicast algo.
Li et al.	Neither	Minimum energy	Source routing	RC
QMRPCAH	Broadcast	QoS	Bordercast	RT
Wu and Jia	Neither	SP and least cost	Source routing	RC
Randaccio and Atzori	Neither	SP, WBW & delay	Source routing	RC
Fireworks	Both	SP, Cohesiveness	Multicast & broadcast	RC
PPMA	Core	Distance & link cost	Source routing	RC
ALMA	Neither	Link breaks	Tree based	RT
AQM	Core	QoS	Source routing	RT
CBM	Core	Threat arrival	Limited broadcast	RC
DDM	–	SP	Source routing	–
ROMANT	Core	Connectivity	Source routing	RT
EraMobile	–	Randomly selected	Local broadcast	–
DGR	Core	SP	Limited flooding	RC
GAMER	Core	SP	Source routing	RC
GeoGrid	Core	Hop count	Flooding or ticket based	None
GeoTora	Broadcast	SP	Limited flooding	RT

Route metric: SP = Shortest path; WBW = Weighted bandwidth

Route repository: RT = Routing table; RC = Route cache

- *Minimize energy consumed per packet*: Let us consider a packet j traversing a path formed of the nodes $n_1, n_2, \ldots, n_k$. Let us denote the energy needed to transfer the packet from node a to b with $T(a, b)$. The goal is to minimize $\sum_{i=1}^{k-1} T(n_i, n_{i+1})$ for all packets j.

- *Maximize time for network partitioning*: This requirement tries to balance the load on the network in such a way as to delay, as much as possible, the moment in which node failures leave the network as two or more disjoint graphs, without interconnections.

- *Minimize variance in node power levels*: This metric directly relates to the extension of the network's life time. No node should be penalized with a higher energy consumption than its peers, assuring that all nodes will remain up and running together for as long as possible.

- *Minimize cost per packet*: The minimization of energy consumption might lead to situations where some nodes in specific locations will need to forward more traffic, exhausting their energy supplies. To extend the life time of the network we need to define a cost metric which can ensure that no node with low energy level will participate too often in routes. We can define a function $f_i(x_i)$ that estimates the "node cost" or "weight" of a node i (where x_i is the total energy expended by i). This intuitively defines the *node's reluctance to forward packets*. Possible forms of this function include combinations of inverse relations between the current battery life and the cost at that node (i.e., higher the battery life, lower the cost at that node).

- Minimize maximum node cost: If $C_i(t)$ denotes the cost to route a packet through node i at time t, and $C'(t)$ denotes the maximum $C(t)$ over all the nodes, the goal is to minimize $C'(t)$. In other words, the aim is to minimize the maximum node cost after routing N packets to their destinations or after T seconds. Using this metric will delay the time point where the first node in the network will fail and will implicitly reduce the variance in the node power levels.

The paper shows through simulations that the pursuit of these optimization criteria is possible and significant cost reductions can be achieved. Naturally, one cannot optimize simultaneously along all axes, as some of the criteria conflict with each other (for instance, for most networks, the minimum energy consumption routes are not the one with the lowest variance on the node energy usage). The paper does not propose full fledged routing protocols, but offer a guidance with respect to what sort of metrics future protocols would need to strive to, and how these would be evaluated.

More than a decade later, we can, with some confidence, say that the authors were very optimistic when stating that the integration of these metrics in routing protocols will be easy. Many subsequent papers in power aware ad hoc routing can be seen as attempts to find practical solutions to the challenges outlined here.

Device and Energy Aware Routing (DEAR) [7]: Avudainayagam et al. propose a protocol for a heterogeneous network where some nodes are on battery power while other nodes are connected to a continuous supply of power (or can be periodically recharged as needed). The goal of the protocol is to rely on the latter type of nodes for most of the routing functionality, thus extending the lifetime of the battery powered nodes.

The DEAR procol calls a node *device-aware* if it can distinguish between whether it is battery powered or externally powered and the cost of using a device of the latter type is zero. The routing table of device-aware nodes has an additional field called *DeviceType* which indicates whether the destination node is on battery power or is externally powered. An additional *redirect table* contains pairs of destination addresses and redirect addresses.

After every routing table update, the node calculates the least cost to any externally powered device from its routing table and updates the redirect table accordingly as well. During routing, the node compares the cost of reaching the destination versus the cost of reaching a powered node, and if the latter is cheaper, it will redirect the packet to a powered node. The DEAR protocol assumes that any powered node can boost its transmission power such that it is at a one-hop distance to any destination, thus the transmission from the powered node is considered to be of zero cost from an energy consumption point of view.

Routing and channel assignment for low power transmission in PCS [107]: Scott and Bombos propose a technique for minimizing the transmission power in PCS networks by the simultaneous choice of the route and channel assignment (this combined problem is called "call placement"). The aim is to

increase the lifetimes of the individual nodes, and hence the network lifetime. The approach chosen is similar to the frequency reuse factor in AMPS cellular service.

Due to the inherent complexity and the overhead involved with the continuous optimization of the entire network, the authors use an approach which is triggered only when new calls are made. In this *least disturbance* approach, new calls are placed such that the total power required to sustain the new call is minimal. This will minimize the new call's interference with the rest of the network.

Energy conserving routing in wireless ad hoc networks [26]: Chang and Tassiulas start with the observation that in general, the routes with the lowest energy cost are the routes with the least hops. It would appear that calculating the shortest path would also minimize the energy use. However, this technique leads to high energy consumption of the nodes which are along the shortest paths, while the battery power of the other nodes in the network remain largely unutilized. Chang and Tassiulas introduce a routing strategy to maximize the network lifetime based on sets of source-destination pairs and the traffic generation rates on these flows. A class of flow augmentation and flow redirection algorithms are proposed which balance energy consumption rates in relation to the energy resources of the nodes. Simulation results show 60% increase of the system lifetime compared to the minimum transmitted energy routing algorithm.

CLUSTERPOW and MINPOW [60]: Kawadia and Kumar designed a series of energy aware algorithms specifically targetting non-homogeneous ad hoc networks. The authors notice that the power control and routing is mutually dependent of each other; for instance routing must be aware of the power control since link quality mainly depends on the available battery power at each node.

The CLUSTERPOW protocol uses a dynamic and implicit clustering of nodes in a network. The clustering criteria is the power levels of each node (in contrast to the traditional clustering approaches based on addresses or geographical placement of nodes). A route is selected by ensuring that each hop in the route has a maximum transmit power level.

The tunneled CLUSTERPOW scheme augments this protocol with the ability to perform packet encapsulation at low power levels instead of sending the packet directly to the next hop.

Finally, the MINPOW protocol relies on the distributed Bellman-Ford algorithm with sequence numbers, using the total power consumption as a metric instead of hop count. To calculate the shortest path, any of the existing shortest path algorithm can be used to calculate the smallest transmit power required to traverse the link. MINPOW takes into consideration the total power consumed in communication instead of individual power levels.

Interference aware cooperative routing [77]: Mahmood and Comaniciu propose an algorithm specifically targeted to CDMA-based ad hoc sensor (in contrast to the collision-based physical layer models assumed by most other algorithms).

CDMA networks almost always implement a variable transmission power to prevent the "nearfar effect", where the useful signal is drowned by strong interference from adjacent transmitters.

The two algorithms proposed by Mahmood and Comaniciu maximize the throughput and minimize energy consumption in ad hoc networks. The proposed strategies try to mitigate the near-far effect at the network layer by making appropriate selections of routes within the network instead of using more sophisticated power control techniques.

The first approach is based on minimum energy routing with additional selection criterion based on the potential interference levels. Once a route has been selected, the algorithm estimates the interference created by each node on the route to neighboring nodes. If this level is higher than a threshold, the node, as a potential near-far effect hot spot, is removed from the route.

The second approach is based on joint minimum energy and near-far minimization routing, using a composite metric. Several additive and multiplicative metrics are proposed and evaluated.

Simulation results show an increase in net throughput of the network of up to 60 percent in comparison with the minimum energy routing.

Minimum Energy Hierarchical Dynamic Source Routing (MEHDSR) [115]: Tarique and Tepe extend the well-known DSR protocol with two protocols which consider energy consideration.

The first protocol, minimum energy dynamic source routing (MEDSR), extends the well known DSR protocol by modifying the control messages and using a link-by-link power adjustment strategy. Source nodes trying to find a new path to the destination first try to find one traversing exlusively links requiring

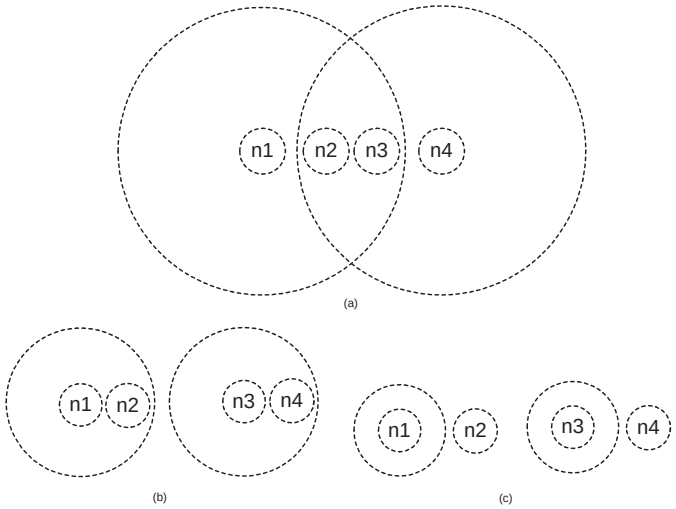


Figure 35: (a) Minimum transmit power, (b) high transmit power, (c) low transmit power [115].

a low power level. If the attempt is unsuccessful, successively higher power levels can be found. The different power levels are used to identify paths which can return low energy routes (see Figure 35). Multiple power levels also reduce the route discovery time and overhead. Once the route discovery process is successful, the transmission power levels of these nodes are adjusted on a link-by-link basis to the minimum required level for each link.

Although MEDSR is creates highly efficient routes (a 25% improvement in energy consumption has been measured), the flooding technique used for route discovery has both a large of overhead and is energy inefficient. The second proposed protocol, HMEDSR, adopts a hierarchical routing approach similar to Hierarchical Dynamic Source Routing (HDSR). The reduction in the routing overhead packets provides an additional 12% energy improvement.

On-line disjoint path routing for capacity maximization [71]: Liang and Liu aim to maximize the capacity of an ad hoc network, defined as the number of successfully routed messages. The assumption is that the algorithms have no knowledge of future path requests or the traffic on those paths (as a note, this is the normal operating condition in ad hoc networks). They are considering a network where the transmission power of every node can be adjusted within the limit of a maximum transmission power. Thus, links can be added or removed to the topology by allowing the nodes to vary their transmission power. In this setting, the routing problem requires both the choice of the nodes on the path as well as the selection of their transmission power.

The authors are proposing two centralized on-line algorithms for the problem. The first algorithm is based on maximizing the local network lifetime thereby minimizing the transmission energy consumption throughout the network. The second algorithm is a heuristic solution based on an exponential function of the energy utilization at the nodes. The minimum energy between two node/edge-disjoint paths are computed and their sum compared with a threshold to determine whether the connection request is accepted or not.

Power conserving routing with entropy-constrained algorithms [58]: Karayiannis and Nadella develop a routing algorithm which utilizes the information-theoretic concept of *entropy* aiming to reduce the uncertainty associated with route discovery.

The paper starts from the idea of entropy constrained routing algorithms, which have been introduced by the authors in several preceeding papers as a mean to implement routing based on multiple performance criteria. The idea is that the optimization criteria is implemented as a series of constraints on the chosen route. In an entropy constrained routing, the initial iteration of the route starts with a level of uncertainty: multiple possible nodes can compete for each position of the route. This uncertainty is gradually reduced by a deterministic annealing process.

In [58] the authors apply this approach to the issue of power conserving routing. As the entropy constrained routing can possibly optimize for more than one performance criteria, two specific imple-

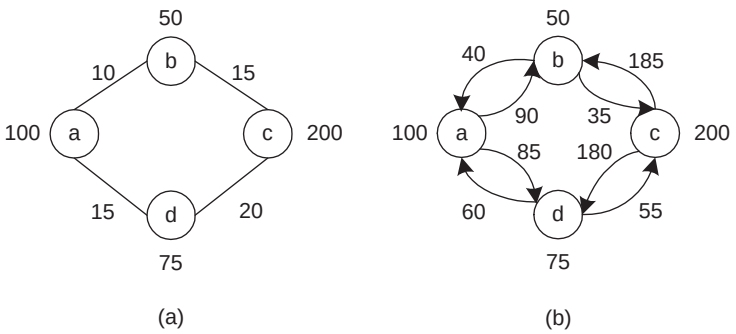


Figure 36: (a) A graph showing energy levels at nodes and energy required to transmit at each edge. (b) shows the corresponding energy graph [81].

mentations are considered:

- optimizing for a single performance metric, with the metric chosen to be the link cost
- multiple performance metrics: the metrics chosen are link cost together with link reliability.

The authors are comparing the performance of the algorithms against traditional approaches using a Dijkstra's algorithm in a mobile ad hoc network and conclude that entropy-constrained algorithms can increase the lifetime of the network (as measured by the time to the first node exhausting its energy).

Online energy aware routing [81]: Mohanoor et al. propose polynomial time combinatorial techniques to compute energy efficient routes in MANETs. The aim is to select a route which strikes a balance between the *residual energy*, the minimum energy level of any node in the path and the *energy consumed* along a path. The network is considered as a graph with the edge weights being the energy required to transmit (see Figure 36(a)). The sum of all weights along a path corresponds to the total energy consumed along the path (see Figure 36(b)).

The authors consider several algorithms, united by their reliance on a two phase computation. The first technique, called the *shortest widest path* works as follows. First, a variant of Dijstra's algorithm is applied to the graph constructed as above, which searches for paths with the minimum residual energy (the "widest" paths). There can be several paths which satisfy this condition. In the second phase, the algorithm chooses from these paths the one which has the lowest energy consumption.

Starting from this approach, the paper also describes several variations on the shortest widest path approach. The *shortest width constrained path* algorithm allows the user to trade off the width of the path for a lower energy consumption. The used can specify the trade-off factor *eta* which describes the acceptable trade-off ratio. The second variation, the *shortest fixed width path* finds the minimum energy path on the path which has a higher width than a fixed value.

The authors show through simulation results that the use of this class of power aware routing algorithms can result in significant improvements in the network lifetime.

3.9.1. Comparison

Power aware routing makes the decision of routes dependent on power consumption characteristics. In general, the goal is to find routes for a collection of well specified flows. Thus, the routes need to be set up on-demand. Pre-emptive routing algorithms can be used only in conjunction with an estimate of future flows.

This makes routing a multi-criteria optimization problem. For the papers surveyed, the number of criteria can be usually considered two: one traffic related (shortest path, bandwidth) and one energy related criteria. A possible exception is the paper by Mahmood and Comaniciu [77] which adds interference to the minimization criteria (although interference, in this case, is strongly tied to bandwidth).

One of the problems is that power-awareness can rarely be compressed into a neat optimization criteria. The absolute amount of power consumed network-wide is almost completely irrelevant, in fact,

Table 8: Power aware routing protocols comparison

Protocol	Type	Path strategy	Routing metric	Scalability	Robustness
DEAR	G	–	Based on "DeviceType"	No	Yes
Scott & Bombos	C	Single-path	Multiple constrained SP	Yes	No
Chang & Tassiulas	G	Single-path	Power cost	No	No
CLUSTERPOW	Cl	Shortest path	Total consumed power	Yes	Yes
Mahmood & Comaniciu	D	Single-path	Energy and interference	No	No
MEHDSR	G	Single-path	SP or next available	Yes	No
Liang & Liu	D	Single-path	Energy consumption	Yes	Yes
Karayiannis & Nadella	D	Single path	Entropy	Yes	No
Mohanoor et al.	D	Single-path	Constrained	SP	No

Type: C = Centralized; Cl = Clustered; D = Distributed; G = Global

Routing metric: SP = Shortest path

none of the routing algorithms discussed here optimizes for it. One reason for this is that the minimization of energy consumption is quite often can be achieved by shortest path routing.

The challenges instead are not simply to minimize the energy consumption, but to manage it. The routing algorithm can redistribute the routes over the network, such that the overall power consumption is redistributed over the network. As the paper of Singh et al. [109] argued in the late 1990's, this cost redistribution can be specified in several alternative ways. In the majority of papers, the ultimate objective is to extend the lifetime of the network. There are two dominant definitions for this. Some papers consider the moment when the first node in the network fails such as Karayiannis and Nadella [58], while most of them consider the moment when the network looses connectivity Chang and Tassiulas [26], Liang and Liu [71], Mohanoor et al. [81].

Another aspect is whether the algorithm is implemented in a centralized [107, 71, 58, 81] or a distributed way [7, 60, 115].

Similarly to QoS routing, one of the difficulties of energy efficient routing is that the optimal route for one particular flow of data cannot be determined, except by considering other currently ongoing paths. Furthermore, theoretically, optimal allocation cannot be achieved by allocating every new flow of data as they are started: we might possibly need to re-route the existing routes, such that a new optimum can be reached. Such a re-routing of existing flows can lead to a better overall route, but it also creates many problems. Thus some of the considered algorithms are explicitly designed to minimize the disturbances, e.g. Scott and Bombos [107], while many others are designed to operate in an online manner - that is, consider the new data flows as they appear.

Another cross-cutting consideration concerns node heterogeneity. Considering this is tricky because different authors mean different things by heterogeneity. First of all, even if we start with perfectly identical nodes, after a while, they can become heterogeneous because of different locations in the network, and implicitly, different battery consumption. Some papers consider hard heterogeneity, where the nodes are physically different, either by their energy resources, or by their transmission range: for instance, DEAR[7].

The paper by Liang and Liu [71] operates with an interesting assumption: it assumes that any node can dynamically extend its transmission power to cover the entire area.

A number of other comparison terms are summarized in Table 3.9.1.

4. Conclusions

In this paper, we introduced a taxonomy of ad hoc routing protocols. We have divided the ad hoc routing protocols into nine categories: i) source-initiated (reactive or on-demand), ii) table-driven (pro-active), iii) hybrid, iv) location-aware (geographical), v) multipath, vi) multicast, vii) geographical multicast, viii) hierarchical, and ix) power-aware. For each of these classes, we reviewed and compared several representative protocols. While different classes of protocol operate under different scenarios, they usually share the common goal to reduce control packet overhead, maximize throughput, and minimize

the end-to-end delay. The main differentiating factor between the protocols is the ways of finding and/or maintaining the routes between source-destination pairs.

The development of the ad hoc routing protocols over the last 15 years is an example of one of the most systematic explorations of a design space in the history of computer science. Although, clearly, newer protocols have built upon the earlier ones, we cannot identify a single "best" protocol. Almost all the protocols we discussed in this paper have their own sweet spot deployment scenarios and performance metric combinations where they outperform their competitors.

From the point of view of the practitioner, this creates a serious problem. To deploy an ad hoc network with an optimal performance, it requires a very careful analysis of the scenario and its requirements, and the appropriate choice of the routing protocol from the dozens applicable in the context. We hope that the taxonomy presented in this paper will be a helpful instrument for making this decision.

References

[1] J.-D. Abdulai, M. Ould-Khaoua, and L. Mackenzie. Adjusted probabilistic route discovery in mobile ad hoc networks. *Computers and Electrical Engineering*, 35(1):168–182, January 2009.

[2] G. N. Aggelou and R. Tafazolli. RDMAR: A bandwidth-efficient routing protocol for mobile ad hoc networks. In *Proceedings of ACM WOWMOM*, pages 26–33, August 1999.

[3] C. Ahn. Gathering-based routing protocol in mobile ad hoc networks. *Computer Communications*, 30(1):202–206, 2006.

[4] J. Al-Karaki and A. Kamal. Efficient virtual-backbone routing in mobile ad hoc networks. *Computer Networks*, 52(2):327–350, 2008.

[5] B. An and S. Papavassiliou. A mobility-based clustering approach to support mobility management and multicast routing in mobile ad-hoc wireless networks. *International Journal of Network Management*, 11(6):387–395, 2001.

[6] B. An and S. Papavassiliou. Geomulticast: architectures and protocols for mobile ad hoc wireless networks. *Journal of Parallel Distributed Computing*, 63(2):182–195, 2003.

[7] A. Avudainayagam, W. Lou, and Y. Fang. Dear: A device and energy aware routing protocol for heterogeneous ad hoc networks. *Journal of Parallel Distributed Computing*, 63(2):228–236, 2003.

[8] R. Bagrodia, M. Gerla, L. Kleinrock, J. Short, and T. Tsai. A hierarchical simulation environment for mobile wireless networks. Technical report, Dept. of Computer Science, University of California at Los Angeles, 1996.

[9] A. Bamis, A. Boukerche, I. Chatzigiannakis, and S. Nikoletseas. A mobility aware protocol synthesis for efficient routing in ad hoc mobile networks. *Computer Networks*, 52(1):130–154, 2008.

[10] S. Basagni. Distributed clustering for ad hoc networks. In *Proceedings of IEEE ISPAN*, pages 310–315, June 1999.

[11] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward. A distance routing effect algorithm for mobility (DREAM). In *Proceedings of the ACM MOBICOM*, pages 76–84, October 1998.

[12] R. Beraldi. The polarized gossip protocol for path discovery in manets. *Ad Hoc Networks*, 6(1):79–91, 2008.

[13] R. Beraldi, L. Querzoni, and R. Baldoni. A hint-based probabilistic protocol for unicast communications in MANETs. *Ad Hoc Networks*, 4(5):547–566, 2006.

[14] L. Blazevic, J.-Y. L. Boudec, and S. Giordano. A location-based routing method for mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 4(2):97–110, 2005.

[15] L. Blazevic, S. Giordano, and J.-Y. L. Boudec. Self organized routing in wide area mobile ad-hoc networks. In *Proceedings of IEEE GLOBECOM*, pages 2814–2818, November 2001.

[16] G. Boato and F. Granelli. MORA: a movement-based algorithm for ad hoc networks. In *Proceedings of The Tenth International Conference on Distributed Multimedia Systems (DMS)*, pages 171–174, 2004.

[17] J. Boice, J. Garcia-Luna-Aceves, and K. Obraczka. Combining on-demand and opportunistic routing for intermittently connected networks. *Ad Hoc Networks*, 7(1):201–218, 2009.

[18] J. Boleng. *Exploiting location information and enabling adaptive mobile ad hoc networking protocols*. PhD thesis, Colorado School of Mines, 2002.

[19] J. Boleng and T. Camp. Adaptive location aided mobile ad hoc network routing. In *Proceedings of IEEE International Performance, Computing, and Communications Conference (IPCCC)*, pages 423–432, April 2004.

[20] A. Boukerche, S. K. Das, and 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):967–995, 2001.

[21] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceedings of ACM MOBICOM*, pages 85–97, October 1998.

[22] K. Bür and C. Ersoy. Ad hoc quality of service multicast routing. *Computer Communications*, 29(1):136–148, 2005.

[23] W. L. C. Chiang, H. Wu and M. Gerla. Routing in clustered multihop, mobile wireless networks. In *Proceedings of IEEE SICON*, pages 197–211, April 1997.

[24] T. Camp and Y. Liu. An adaptive mesh-based protocol for geocast routing. *Journal of Parallel Distributed Computing*, 63(2):196–213, 2003.

[25] N. Carlsson and D. Eager. Non-euclidian geographic routing in wireless networks. *Ad Hoc Networks*, 5(7):1173–1193, 2007.

[26] J.-H. Chang and L. Tassiulas. Energy conserving routing in wireless ad-hoc networks. In *Proceedings of IEEE INFOCOM*, pages 22–31, March 2000.

[27] T.-W. Chen and M. Gerla. Global state routing: a new routing scheme for ad-hoc wireless networks. In *Proceedings of IEEE ICC*, volume 1, pages 171–175, June 1998.

[28] Z. Cheng and W. Heinzelman. Discovering long lifetime routes in mobile ad hoc networks. *Ad Hoc Networks*, 6(5):661–674, 2008.

[29] C.-C. Chiang and M. Gerla. On-demand multicast in mobile wireless networks. In *Proceedings of IEEE ICNP*, pages 262–270, October 1998.

[30] C.-H. Chou, K.-F. Ssu, and H. Jiau. Dynamic route maintenance for geographic forwarding in mobile ad hoc networks. *Computer Networks*, 52(2):418–431, 2008.
[31] T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, A. Qayyum, and L. Viennot. Optimized link state routing protocol for ad hoc networks. In *Proceedings of IEEE INMIC*, pages 62–68, December 2001.
[32] M. Colagrosso, N. Enochs, and T. Camp. Improvements to location-aided routing through directional count restrictions. In *Proceedings of International Conference on Wireless Networks (ICWN)*, pages 924–929, June 2004.
[33] M. Conti, E. Gregori, and G. Maselli. Reliable and efficient forwarding in ad hoc networks. *Ad Hoc Networks*, 4(3):398–415, 2006.
[34] M. S. Corson and A. Ephremides. A distributed routing algorithm for mobile wireless networks. *ACM/Baltzer Wireless Networks Journal*, 1(1):61–81, February 1995.
[35] S. K. Das, B. S. Manoj, and C. S. R. Murthy. A dynamic core based multicast routing protocol for ad hoc wireless networks. In *Proceedings of ACM MobiHoc*, pages 24–35, June 2002.
[36] S. K. Das, A. Mukherjee, S. Bandyopadhyay, D. Saha, and K. Paul. An adaptive framework for QoS routing through multiple paths in ad hoc wireless networks. *Journal of Parallel Distributed Computing*, 63(2):141–153, 2003.
[37] R. Dube, C. D. Rais, K. Y. Wang, and S. K. Tripathi. Signal stability-based adaptive routing (SSA) for ad hoc mobile networks. *IEEE Personal Communications Magazine*, pages 36–45, February 1997.
[38] J. Eisbrener, G. Murphy, D. Eade, C. Pinnow, K. Begum, S. Park, S.-M. Yoo, and J.-H. Youn. Recycled path routing in mobile ad hoc networks. *Computer Communications*, 29(9):1552–1560, 2006.
[39] J. Eriksson, M. Faloutsos, and S. V. Krishnamurthy. Scalable ad hoc routing: The case for dynamic addressing. In *Proceedings of IEEE INFOCOM*, volume 2, pages 1108–1119, March 2004.
[40] J. Garcia-Luna-Aceves, M. Mosko, and C. Perkins. A new approach to on-demand loop-free routing in networks using sequence numbers. *Computer Networks*, 50(10):1599–1615, 2006.
[41] J. J. Garcia-Luna-Aceves. Loop-free routing using diffusing computations. *IEEE/ACM Transactions on Networking*, 1(1):130–141, 1993.
[42] J. J. Garcia-Luna-Aceves and E. L. Madruga. A multicast routing protocol for ad-hoc networks. In *Proceedings of IEEE INFOCOM*, pages 784–792, March 1999.
[43] J. J. Garcia-Luna-Aceves and M. Spohn. Source-tree routing in wireless networks. In *Proceedings of IEEE ICNP*, pages 273–282, October-November 1999.
[44] M. Ge, S. Krishnamurthy, and M. Faloutsos. Application versus network layer multicasting in ad hoc networks: the ALMA routing protocol. *Ad Hoc Networks*, 4(2):283–300, 2006.
[45] J. Ghosh, S. Philip, and C. Qiao. Sociological orbit aware location approximation and routing (solar) in manet. *Ad Hoc Networks*, 5(2):189–209, 2007.
[46] V. Giruka and M. Singhal. A self-healing on-demand geographic path routing protocol for mobile ad-hoc networks. *Ad Hoc Networks*, 5(7):1113–1128, 2007.
[47] T. Goff, N. B. Abu-Ghazaleh, D. S. Phatak, and R. Kahvecioglu. Preemptive routing in ad hoc networks. *Journal of Parallel Distributed Computing*, 63(2):123–140, 2003.
[48] M. Gong, S. Midkiff, and S. Mao. On-demand routing and channel assignment in multi-channel mobile ad hoc networks. *Ad Hoc Networks*, 7(1):63–78, 2009.
[49] C. Gui and P. Mohapatra. Hierarchical multicast techniques and scalability in mobile ad hoc networks. *Ad Hoc Networks*, 4(5):586–606, 2006.
[50] M. Gunes, U. Sorges, and I. Bouazizi. Ara–the ant-colony based routing algorithms for manets. In *Proceedings of IEEE ICPP Workshop on Ad Hoc Networks (IWAHN)*, pages 79–85, August 2002.
[51] Z. J. Haas. A new routing protocol for the reconfigurable wireless networks. In *Proceedings of IEEE ICUPC*, pages 562–566, October 1997.
[52] G. Ivascu, S. Pierre, and A. Quintero. QoS routing with traffic distribution in mobile ad hoc networks. *Computer Communications*, 32(2):305–316, February 2009.
[53] A. Iwata, C. Chiang, G. Pei, M. Gerla, and T. Chen. Scalable routing strategies for ad hoc wireless networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1369–1379, 1999.
[54] J. Jetcheva and D. Johnson. Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks. In *Proceedings of ACM MobiHoc*, pages 33–44, October 2001.
[55] L. Ji and M. Corson. Differential destination multicast-a manet multicast routing protocol for small groups. *Proceedings of IEEE INFOCOM*, 2:1192–1201, April 2001.
[56] M. Joa-Ng and I.-T. Lu. A peer-to-peer zone-based two-level link state routing for mobile ad hoc network. *IEEE Journal on Selected Areas in Communications*, 17(8):1415–1425, 1999.
[57] D. B. Johnson, D. A. Maltz, and J. Broch. DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks. In C. E. Perkins, editor, *Ad Hoc Networking*, chapter 5, pages 139–172. Addison-Wesley, 2001.
[58] N. Karayiannis and S. Nadella. Power-conserving routing of ad hoc mobile wireless networks based on entropy-constrained algorithms. *Ad Hoc Networks*, 4(1):24–35, 2006.
[59] B. Karp and H. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In *Proceedings of ACM MobiCom*, pages 243–254, August 2000.
[60] V. Kawadia and P. Kumar. Power control and clustering in ad hoc networks. In *Proceedings of IEEE INFOCOM*, volume 1, pages 459–469, March-April 2003.
[61] Y. Ko and N. Vaidya. Geocasting in mobile ad hoc networks: Location-based multicast algorithms. Technical report, TR-98-018, Texas AM University, September 1998.
[62] Y. Ko and N. Vaidya. Location-Aided Routing (LAR) in mobile ad hoc networks. In *Proceedings of ACM MobiCom*, pages 66–75, October 1998.
[63] Y.-B. Ko and N. H. Vaidya. GeoTORA: A protocol for geocasting in mobile ad hoc networks. In *Proceedings of IEEE ICNP*, pages 240–249, November 2000.
[64] S. Kwon and N. Shroff. Geographic routing in the presence of location errors. *Computer Networks*, 50(15):2902–2917, 2006.
[65] W. Lai, S.-Y. Hsiao, and Y.-C. Lin. Adaptive backup routing for ad-hoc networks. *Computer Communications*, 30(2):453–464, 2007.

[66] L. Law, S. Krishnamurthy, and M. Faloutsos. A novel adaptive protocol for lightweight efficient multicasting in ad hoc networks. *Computer Networks*, 51(3):823–834, 2007.
[67] L. Layuan and L. Chunlin. A QoS multicast routing protocol for clustering mobile ad hoc networks. *Computer Communications*, 30(7):1641–1654, 2007.
[68] S. Lee and M. Gerla. Split multipath routing with maximally disjoint paths in ad hoc networks. In *Proceedings of the IEEE ICC*, pages 3201–3205, June 2001.
[69] D. Li, Q. Liu, X. Hu, and X. Jia. Energy efficient multicast routing in ad hoc wireless networks. *Computer Communications*, 30(18):3746–3756, 2007.
[70] J. Li and P. Mohapatra. LAKER: Location aided knowledge extraction routing for mobile ad hoc networks. In *Proceedings of IEEE WCNC*, pages 1180–1184, March 2003.
[71] W. Liang and Y. Liu. On-line disjoint path routing for network capacity maximization in energy-constrained ad hoc networks. *Ad Hoc Networks*, 5(2):272–285, 2007.
[72] W.-H. Liao, J.-P. Sheu, and Y.-C. Tseng. GRID: A fully location-aware routing protocol for mobile ad hoc networks. *Telecommunication Systems*, 18(1-3):37–60, 2001.
[73] W.-H. Liao, Y. Tseng, K. Lo, and J. Sheu. GeoGRID: A geocasting protocol for mobile ad hoc networks based on grid. *Journal of Internet Technology*, 1(2):23–32, 2000.
[74] C. Liu, M. Yarvis, W. Conner, and X. Guo. Guaranteed on-demand discovery of node-disjoint paths in ad hoc networks. *Computer Communications*, 30(14-15):2917–2930, 2007.
[75] J.-S. Liu and C.-H. R. Lin. RBR: refinement-based route maintenance protocol in wireless ad hoc networks. *Computer Communications*, 28(8):908–920, 2005.
[76] Y. Liu, X. Hu, M. Lee, and T. Saadawi. A region-based routing protocol for wireless mobile ad hoc networks. *IEEE Network*, 18(4):12–17, 2004.
[77] H. Mahmood and C. Comaniciu. Interference aware cooperative routing for wireless ad hoc networks. *Ad Hoc Networks*, 7(1):248–263, 2009.
[78] C. Maihöfer. A survey on geocast routing protocols. *IEEE Communications Surveys and Tutorials*, 6(2):32–42, 2004.
[79] M. K. Marina and S. Das. On-demand multi path distance vector routing in ad hoc networks. In *Proceedings of IEEE ICNP*, pages 14–23, November 2001.
[80] R. Mavropodi, P. Kotzanikolaou, and C. Douligeris. SecMR - a secure multipath routing protocol for ad hoc networks. *Ad Hoc Networks*, 5(1):87–99, 2007.
[81] A. Mohanoor, S. Radhakrishnan, and V. Sarangan. Online energy aware routing in wireless networks. *Ad Hoc Networks*, 7(5):918–931, 2009.
[82] A. Munaretto and M. Fonseca. Routing and quality of service support for mobile ad hoc networks. *Computer Networks*, 51(11):3142–3156, 2007.
[83] S. Murthy and J. J. Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. *MONET*, 1(2):183–197, 1996.
[84] J. Na and C.-K. Kim. Glr: A novel geographic routing scheme for large wireless ad hoc networks. *Computer Networks*, 50(17):3434–3448, 2006.
[85] M. Naserian and K. Tepe. Game theoretic approach in routing protocol for wireless ad hoc networks. *Ad Hoc Networks*, 7(3):569–578, 2009.
[86] N. Nikaein and C. Bonnet. HARP:hybrid ad hoc routing protocol. In *Proceedings of International Symposium on Telecommunications (IST)*, pages 56–67, 2001.
[87] N. Nikaein, H. Labiod, and C. Bonnet. DDR: distributed dynamic routing algorithm for mobile ad hoc networks. In *Proceedings of ACM MobiHoc*, pages 19–27, August 2000.
[88] O. Ozkasap, Z. Genc, and E. Atsan. Epidemic-based reliable and adaptive multicast for mobile ad hoc networks. *Computer Networks*, 53(9):1409–1430, 2009.
[89] V. D. Park and M. S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proceedings of INFOCOM*, pages 1405–1413, April 1997.
[90] G. Pei, M. Gerla, and T.-W. Chen. Fisheye state routing in mobile ad hoc networks. In *Proceedings of IEEE ICDCS Workshop on Wireless Networks and Mobile Computing*, pages D71–D78, April 2000.
[91] G. Pei, M. Gerla, and X. Hong. LANMAR: Landmark routing for large scale wireless ad hoc networks with group mobility. In *Proceedings of ACM MobiHoc*, pages 11–18, August 2000.
[92] C. Perkins, editor. *Ad hoc Networking*. Addison Wesley, 2001.
[93] C. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *ACM SIGCOMM*, pages 234–244, August-September 1994.
[94] C. Perkins and E. Royer. Ad hoc On-demand Distance Vector Routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 99–100, 1999.
[95] D. Pompili and M. Vittucci. PPMA, a probabilistic predictive multicast algorithm for ad hoc networks. *Ad Hoc Networks*, 4(6):724–748, 2006.
[96] S. Radhakrishnan, N. Rao, G. Racherla, C. Sekharan, and S. Batsell. DST–a routing protocol for ad hoc networks using distributed spanning trees. In *Proceedings of IEEE WCNC*, pages 100–104, September 1999.
[97] S. Rajagopalan and C.-C. Shen. ANSI: A swarm intelligence-based unicast routing protocol for hybrid ad hoc networks. *Journal of Systems Architecture*, 52(8-9):485–504, 2006.
[98] J. Raju and J. Garcia-Luna-Aceves. A new approach to on-demand loop-free multipath routing. In *Proceedings of the Eight IEEE International Conference on Computer Communications and Networks (IC3N)*, pages 522–527, April 1999.
[99] S. Ramasubramanian, H. Krishnamoorthy, and M. Krunz. Disjoint multipath routing using colored trees. *Computer Networks*, 51(8):2163–2180, 2007.
[100] L. Randaccio and L. Atzori. Group multicast routing problem: A genetic algorithms based approach. *Computer Networks*, 51(14):3989–4004, 2007.
[101] H. Rangarajan and J. Garcia-Luna-Aceves. Efficient use of route requests for loop-free on-demand routing in ad hoc networks. *Computer Networks*, 51(6):1515–1529, 2007.
[102] L. R. Reddy and S. Raghavan. SMORT: Scalable multipath on-demand routing for mobile ad hoc networks. *Ad Hoc*

Networks, 5(2):162–188, 2007.
[103] T. B. Reddy, S. Sriram, B. Manoj, and C. S. R. Murthy. MuSeQoR: Multi-path failure-tolerant security-aware QoS routing in ad hoc wireless networks. *Computer Networks*, 50(9):1349–1383, 2006.
[104] L. Rosati, M. Berioli, and G. Reali. On ant routing algorithms in ad hoc networks with critical connectivity. *Ad Hoc Networks*, 6(6):827–859, 2008.
[105] S. Roy and J. J. Garcia-Luna-Aceves. Using minimal source trees for on-demand routing in ad hoc networks. In *Proceedings of IEEE INFOCOM*, pages 1172–1181, April 2001.
[106] P. Samar, M. Pearlman, and S. Haas. Independent zone routing: An adaptive hybrid routing framework for ad hoc wireless networks. *IEEE/ACM Transactions on Networking*, 12(4):595–608, August 2004.
[107] K. Scott and N. Bombos. Routing and channel assignment for low power transmission in PCS. In *Proceedings of IEEE ICUPC*, pages 498–502, September-October 1996.
[108] C. Sengul and R. Kravets. Bypass routing: An on-demand local recovery protocol for ad hoc networks. *Ad Hoc Networks*, 4(3):380–397, 2006.
[109] S. Singh, M. Woo, and C. Raghavendra. Power-aware routing in mobile ad hoc networks. In *Proceedings of ACM MobiCom*, pages 181–190, October 1998.
[110] R. Sivakumar, P. Sinha, and V. Bharghavan. CEDAR: a core-extraction distributed ad hoc routing algorithm. *IEEE Journal on Selected Areas in Communications*, 17(8):1454–1465, 1999.
[111] J.-H. Song, V. Wong, and V. Leung. Secure position-based routing protocol for mobile ad hoc networks. *Ad Hoc Networks*, 5(1):76–86, 2007.
[112] O. Souihli, M. Frikha, and M. Hamouda. Load-balancing in MANET shortest-path routing protocols. *Ad Hoc Networks*, 7(2):431–442, 2009.
[113] W. Su and M. Gerla. IPv6 flow handoff in ad-hoc wireless networks using mobility prediction. In *Proceedings of IEEE GLOBECOM*, pages 271–275, December 1999.
[114] B. Sun and L. Li. QoS-aware multicast routing protocol for ad hoc networks. *Journal of Systems Engineering and Electronics*, 17(2):417–422, 2006.
[115] M. Tarique and K. Tepe. Minimum energy hierarchical dynamic source routing for mobile ad hoc networks. *Ad Hoc Networks*, 7(6):1125–1135, 2009.
[116] 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):103–139, March 1997.
[117] R. Vaishampayan and J. Garcia-Luna-Aceves. Efficient and robust multicast routing in mobile ad hoc networks. *Proceedings of IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, pages 304–313, October 2004.
[118] A. Valera, W. K. G. Seah, and S. V. Rao. Cooperative packet caching and shortest multipath routing in mobile ad hoc networks. In *Proceedings of IEEE INFOCOM*, volume 1, pages 260–269, April 2003.
[119] L. Villasenor-Gonzalez, Y. Ge, and L. Lamont. HOLSR: A Hierarchical Proactive Routing Mechanism for Mobile Ad hoc Networks. *IEEE Communications Magazine*, 43(7):118–125, 2005.
[120] G. Wang, Y. Ji, D. C. Marinescu, and 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)*, pages 69–76, October 2004.
[121] G. Wang, D. Turgut, L. Bölöni, Y. Ji, and D. Marinescu. Improving routing performance through m-limited forwarding in power-constrained wireless networks. *Journal of Parallel and Distributed Computing (JPDC)*, 68(4):501–514, April 2008.
[122] J. Wang, E. Osagie, P. Thulasiraman, and R. Thulasiram. Hopnet: A hybrid ant colony optimization routing algorithm for mobile ad hoc network. *Ad Hoc Networks*, 7(4):690–705, 2009.
[123] N.-C. Wang, Y.-F. Huang, and J.-C. Chen. A stable weight-based on-demand routing protocol for mobile ad hoc networks. *Information Sciences*, 177(24):5522–5537, 2007.
[124] Y. Wang, V. Giruka, and M. Singhal. Truthful multipath routing for ad hoc networks with selfish nodes. *Journal of Parallel and Distributed Computing*, 68(6):778–789, 2008.
[125] Y.-H. Wang and C.-F. Chao. Dynamic backup routes routing protocol for mobile ad hoc networks. *Information Sciences*, 176(2):161–185, 2006.
[126] B. Williams and T. Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In *Proceedings of ACM MobiHoc*, pages 194–205, June 2002.
[127] S.-C. Woo and S. Singh. Scalable routing protocol for ad hoc networks. *Wireless Networks*, 7(5):513–529, 2001.
[128] H. Wu and X. Jia. QoS multicast routing by using multiple paths/trees in wireless ad hoc networks. *Ad Hoc Networks*, 5(5):600–612, 2007.
[129] X. Xiaochuan, W. Gang, W. Keping, W. Gang, and J. Shilou. Link reliability based hybrid routing for tactical mobile ad hoc network. *Journal of Systems Engineering and Electronics*, 19(2):259–267, 2008.
[130] J. Xie, R. R. Talpade, A. McAuley, and M. Liu. Amroute: Ad hoc multicast routing protocol. *MONET*, 7(6):429–439, 2002.
[131] K. Xu, X. Hong, and M. Gerla. Landmark routing in ad hoc networks with mobile backbones. *Journal of Parallel and Distributed Computing*, 63(2):110–122, 2003.
[132] Q. Xue and A. Ganz. Ad hoc QoS on-demand routing (AQOR) in mobile ad hoc networks. *Journal of Parallel Distributed Computing*, 63(2):154–165, 2003.
[133] C.-C. Yang and L.-P. Tseng. Fisheye zone routing protocol: A multi-level zone routing protocol for mobile ad hoc networks. *Computer Communications*, 30(2):261–268, 2007.
[134] Z. Yao, J. Jiang, P. Fan, Z. Cao, and V. Li. A neighbor-table-based multipath routing in ad hoc networks. In *Proceedings of IEEE VTC 2003-Spring*, volume 3, pages 1739–1743, April 2003.
[135] C. Yu, T.-K. Wu, and R. Cheng. A low overhead dynamic route repairing mechanism for mobile ad hoc networks. *Computer Communications*, 30(5):1152–1163, 2007.
[136] M. Yu, A. Malvankar, W. Su, and S. Foo. A link availability-based QoS-aware routing protocol for mobile ad hoc sensor networks. *Computer Communications*, 30(18):3823–3831, 2007.
[137] M. Yuksel, R. Pradhan, and S. Kalyanaraman. An implementation framework for trajectory-based routing in ad hoc networks. *Ad Hoc Networks*, 4(1):125–137, 2006.

[138] H. Zhou and S. Singh. Content based multicast (CBM) in ad hoc networks. In *Proceedings of ACM MobiHoc*, pages 51–60, August 2000.
[139] Temporally ordered routing algorithm. Available from web page http://wiki.uni.lu/secan-lab/temporally-ordered+routing+algorithm.html.
[140] A survey of defense technology: The software revolution - to dissolve, to disappear, June 1995.