CHAPTER 2

ROUTING PROTOCOLS FOR MANET

2.1 INTRODUCTION TO AD HOC NETWORKS

A wireless mobile ad hoc network consists of mobile nodes that are interconnected by wireless multi-hop communication paths. These ad hoc wireless networks are self-creating, self-organizing, and self-administering. Figure 2.1 depicts a sample mobile ad hoc network.

![Sample Ad Hoc Network](image)

Figure 2.1 A sample ad hoc network consisting of mobile nodes

These mobile ad hoc networks offer unique benefits and versatility for certain environments and applications. With no prerequisites of fixed infrastructure or base stations, network can be created and used anytime, anywhere. Such networks could be intrinsically fault-resilient, for they do not operate under the limitations of a fixed topology. Since all nodes are allowed to be mobile, the topology of such networks is necessarily time varying. The addition and deletion of nodes occur only by interactions with other nodes. Thus these types of networks offer many advantages where setting up wired-
line networks are not feasible. Such advantages attracted immediate interest in its early use among military, police, and rescue agencies, and especially under disorganized or hostile environments, including isolated scenes of natural disaster and armed conflict.

Recently, home or small-office networking and collaborative computing with laptop computers in a small area (e.g., a conference or classroom, single building, convention center, etc.) have emerged as other major areas of application.

This concept of mobile ad hoc networking emerged as an attempt to extend the services provided by the traditional Internet to the wireless mobile environment. All current works, as well as this research work consider the ad hoc networks as a wireless extension to the Internet, based on the ubiquitous IP networking mechanisms and protocols. Today’s Internet possesses an essentially static infrastructure where network elements are interconnected over traditional wire-line technology, and these elements, especially the elements that provide the routing or switching functions, do not move. In a mobile ad hoc network, by definition, all the network elements move. As a result, numerous more stringent challenges must be overcome to realize the practical benefits of ad hoc networking.

In addition, the mobility of nodes imposes limitations on their power capacity, and on their transmission range. These nodes must often also satisfy stringent weight limitations for portability. Further these mobile hosts are no longer just end systems; they are also required to relay packets generated by other nodes, hence each node must also be able to function as a router. As the nodes move in and out of range with respect to each other, including those that are operating as routers, the resulting topology needs to be communicated to all other nodes as appropriate to maintain the connectivity information. In accommodating the communication needs of the
user applications, the limited bandwidth of wireless channels and their generally-hostile transmission characteristics, impose additional constraints on how much administrative and control information may be exchanged, and how often. Ensuring effective routing is one of the major challenges for ad hoc networking.

When mobile ad hoc networks are considered for complex applications, the various Quality of Service (QoS) attributes for these applications must be satisfied as a set of pre-determined service requirements. In addition, because of the increasing use of the ad hoc networks for military/police use, and also due to the increasing commercial applications being envisioned to be supported on these types of networks, various security issues also need to be addressed.

2.2 AD HOC WIRELESS NETWORK: OPERATING PRINCIPLES

To illustrate the general operating principles of a mobile ad hoc network, Figure 2.2 depicts the peer-level, multi-hop representation of a sample ad hoc network. Here, mobile node A communicates directly (single-hop) with another such node B whenever a radio channel with adequate propagation characteristics is available between them. Otherwise, multi-hop communication is necessary where one or more intermediate nodes must act as a relay (router) between the communicating nodes. For example, there is no direct radio channel (shown by the lines) between A and C or between A and E as shown in Figure 2.2. Nodes B and D must serve as intermediate routers for communication between A and C, and between A and E, respectively. Thus, a distinguishing feature of ad hoc networks is that all nodes must be able to function as routers on demand along with acting as source and destination for packets. To prevent packets from traversing infinitely long
paths, an obvious essential requirement for choosing a path is that it must be loop-free. And this loop-free path between a pair of nodes is called a route.

![Figure 2.2 Example of an ad hoc network](image)

An ad hoc network begins with at least two nodes, broadcasting their presence (beaconing) with their respective address information. If node A is able to establish direct communication with node B as in Figure 2.2, verified by exchanging suitable control messages between them, they both update their routing tables. When a third node C joins the network with its beacon signal, two scenarios are possible. The first is where both A and B determine that single-hop communication with C is feasible. The second is where only one of the nodes, say B, recognizes the beacon signal from C and establishes direct communication with C. The distinct topology updates, consisting of both address and route updates, are made available in all three nodes immediately afterwards. In the first case, all routes are direct. For the other, the route update first happens between B and C, then between B and A, and then again between B and C, confirming the mutual reachability between A and C via B. As the node moves, it may cause the reachability relations to change in time, requiring route updates. Assume that, for some reason, the link between B and C is no longer available as shown in Figure 2.3. Nodes A and C are still reachable from each other, although this time only via nodes D and E. Equivalently, the original loop-free route $A \leftrightarrow B \leftrightarrow C$ is now replaced.
by the new loop-free route $A \leftrightarrow D \leftrightarrow E \leftrightarrow C$. All five nodes in the network are required to update their routing tables appropriately to reflect this topology change, which will be first detected by nodes B and C, then communicated to A, E, and D. This reachability relation among the nodes may also change for various reasons. For example, a node may wander too far out of range, its battery may be depleted, or it may just suffer from software or hardware failure.

![Topology update owing to a link failure](image)

**Figure 2.3 Topology update owing to a link failure**

As more nodes join the network, or some of the existing nodes leave, the topology updates become more numerous, complex, and usually, more frequent, thus diminishing the network resources available for exchanging user information (i.e., data). Finding a loop-free path between a source-destination pair may therefore become impossible if the changes in network topology occur too frequently. Too frequently here means that there may not be enough time to propagate to all the pertinent nodes the changes arising from the last change in network topology. Thus the ability to communicate degrades with increasing mobility and as a result the knowledge of the network topology becomes increasingly inconsistent. A network is combinatorial stable if, and only if, the topology changes occur slowly enough to allow successful propagation of all topology updates as necessary or if the routing algorithm is efficient enough to propagate the changes in the
network before the next change occurs. Clearly, combinatorial stability is
determined not only by the connectivity properties of the networks, but also
by the efficiency of the routing protocol in use and the instantaneous
computational capacity of the nodes, among others. Combinatorial stability
thus forms an essential consideration for attaining efficient routing objectives
in an ad hoc network. In the next section, some of the characteristics that an
efficient routing algorithm for an ad hoc network must posses are listed in
order to increase the throughput of the network with the least amount of
overhead.

2.2.1 Desirable Properties of an Efficient Routing Algorithm

There are some common desirable properties that any routing
protocol for an ad hoc network should possess. These are:

- Loop free: Presence of loops in the path from the source to the
destination result in inefficient routing. In the worst-case
situation, the packets may keep traversing the loop indefinitely
and never reach their destination.

- Distributed control: In a centralized routing scheme, one node
stores all the topological information and makes all routing
decisions; therefore, it is neither robust, nor scalable. The
central router can be a single point of failure; also, the network
in the vicinity of the central router may get congested with
routing queries and responses.

- Fast routing: The quicker the routing decisions are made, the
sooner the packets can be routed towards the destination, as the
probability that the packets take the chosen route before it gets
disrupted because of node mobility is quite high.
• Localized reaction to topological changes: Topological changes in one part of the network should lead to minimal changes in routing strategy in other distant parts of the network. This will keep the routing update overheads in check and make the algorithm scalable.

• Multiplicity of routes: Even if node mobility results in disruption of some routes, other routes should be available for packet delivery.

• Power efficient: A routing protocol should be power efficient. That is the protocol should distribute the load; otherwise shut-off nodes may cause partitioned topologies that may result in inaccessible routes.

• Secure: A routing protocol should be secure. An authentication is needed for communicating nodes, non-repudiation and encryption for private networking to avoid routing deceptions.

• QoS aware: A routing protocol should also be aware of Quality of Service. A routing protocol needs to know about the delay and throughput for a source destination pair, and must be able to verify its longevity so that a real-time application may rely on it.

2.3 ROUTING IN MOBILE AD HOC NETWORKS

So far a general overview of an ad hoc network and its operating principles has been given. In this section a deeper look into the various approaches for routing in ad hoc networks proposed so far is described. Several approaches toward routing in ad hoc networks have been proposed
with the goal of achieving efficient routing with the ever-changing topology. An intuitive approach of routing messages could be that the sender of the message specifies the exact path that the message should take from the sender to the receiver. But this assumes that the sender knows the entire topology of the network, which is not quite a realistic assumption. Another alternative could be to forward the message to a neighbor in the general direction of the destination, and the neighbor then makes a similar decision regarding how to route the message.

Based on the above-mentioned approaches, many routing algorithms have been put forward. And as mentioned earlier, most of the routing algorithms proposed so far for ad hoc networks have been adapted from the routing techniques employed in wire-line networks that have fixed network topology. But the routing algorithms devised for wire-line networks are based on the determination of the shortest path between the source and the destination. Hence these routing algorithms cannot be applied to ad hoc networks without modification as the network topology in ad hoc networks. The changes in topology in one region of the network alter the shortest path between several pairs of nodes in other regions of the network. Also, in ad hoc networks, this information takes time to propagate to other nodes. If the topology information is not updated promptly, the nodes may continue to route messages based on the outdated information, which may lead to packet losses.

2.3.1 Classification of Routing Algorithms for Ad Hoc Networks

Having seen the composition of mobile ad hoc networks, a centralized routing scheme is completely ruled out, as this might lead to a single point of failure. This leads to the requirement of a distributed routing protocol where every node takes part in making routing decision and maintaining the topology by sharing information among them. These
distributed routing protocols can be classified broadly, first by how they intend to determine the topology of the network, and second by when they make a decision to find a route to a destination. The first category of topology intended routing can further be classified into hierarchical and flat routing, and the second category can further be divided into proactive and reactive routing.

2.3.1.1 Hierarchical Routing

In hierarchical routing algorithms, a set of nodes is divided into clusters. Each cluster has a node, which is designated as the cluster-head. So, every node is either a cluster head or one wireless hop away from the cluster head as shown in Figure 2.4. A node that is not a cluster head, but adjacent to more than one cluster head, is referred to as a gateway. Packets between cluster-heads are routed through gateways.

Finally, nodes that are neither cluster heads nor gateways are referred to as ordinary nodes. The subnet comprising the cluster heads and gateways is referred to as the backbone network. Here, each cluster head maintains information about other nodes in its cluster, and from time to time, this information is exchanged between cluster heads over the network. Thus, the cluster heads gather network topology information. A node that has a packet to send to another node can obtain routing information from its cluster head.

It is not necessary for a packet to be routed through the backbone network, as data packets may be routed along other more efficient routes in the network.
This approach is also termed as topology aware routing as the nodes use the knowledge of the network topology to route messages. There are several ways of implementing this approach. The first possibility is that each node determines the optimal path to every node in the system and stores this information. Each time a stream of packets has to be sent from a source to a destination, a connection is established between two end-points and all the packets follow this path. However, with a changing topology, nodes will have to update their routing information and reestablish paths that were broken during communication. If the network topology does not change very often, it is likely that the path establishment costs are incurred once in the beginning and every subsequent packet is routed without additional overhead. The second possibility is connectionless routing, where a route is determined on the fly for every packet as it moves from one node to another. This method will require nodes to store less information about network topology. However, every packet incurs the routing overhead.
2.3.1.2 Flat Routing

In case of flat routing algorithms all nodes act as routers and share the responsibility of forwarding packets destined to other nodes. Thus, there is no need to elect cluster heads and periodically reorganize the network. Most flat routing algorithms try to implement a distributed version of the shortest path algorithm or flooding where the general approach is that the sender sends a copy of the message to every neighbor. The neighbors then propagate copies of the message to all their neighbors except the one from which they received the message. This process is repeated until the network is entirely flooded with the message. If the destination node is in the same partition as the source, the message is sure to reach the destination.

2.3.1.3 Proactive Routing

In proactive routing algorithms, each node maintains a routing table containing the next hop information for every other node in the network, and hence a route between the source node and the destination node is always available making the approach proactive. Examples of proactive protocols include DSDV and the Fisheye State Routing (FSR).

2.3.1.4 Reactive Routing

In reactive routing algorithms, a path discovery process determines the path to the destination only when the node has a packet to forward that is it reacts to a request to send data to a host. These types of routing algorithms are also referred to as on-demand routing protocols. Two prominent examples are Dynamic Source Routing (DSR) and Ad hoc On demand Distance Vector (AODV) routing algorithm.
A broad classification of the various approaches proposed so far is presented, but different authors use different classification schemes for these basic routing functions. Regardless of the approach, routing protocols for ad hoc networks need to perform a set of basic functions in the form of route identification and route reconfiguration. For communication to be possible, at least one route (i.e., a loop-free path) must exist between any node pair. Route identification functions, as the name suggests, identify a route between a pair of nodes. Route reconfiguration functions are invoked to recover from the effects of undesirable events, such as host or link failures of various kinds, and traffic congestion appearing within a sub-network. Evidently, the recognition of changes in the network topology and the topology update functions constitute an important part of the route reconfiguration functions. A separate category of resource management functions are also considered to ensure that all the network resources are available, to the extent possible, in support of some special objectives, such as those associated with QoS or security.

2.3.2 Routing Algorithms

Based on the various approaches toward routing in mobile ad hoc networks described in the last section, a general overview of various routing algorithms proposed in the literature are given below.

2.3.2.1 Destination Sequenced Distance Vector (DSDV)

The Destination Sequence Distance Vector is the best-known protocol for a pro-active routing scheme. It is based on the classical Distributed Bellman Ford (DBF) routing algorithm for wired networks. DSDV is a table driven routing algorithm, like every other table driven routing algorithm, the nodes here maintain routing tables, which provide
information about every possible destination within the network. A sample routing table in DSDV is shown in Table 2.1.

**Table 2.1 Example routing table entry in DSDV**

<table>
<thead>
<tr>
<th>Destination Addr.</th>
<th>Metric (No. of hops)</th>
<th>Addr. of next hop</th>
<th>Sequence no.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host 1</td>
<td>2</td>
<td>host 2</td>
<td>S1</td>
</tr>
<tr>
<td>Host 2</td>
<td>1</td>
<td>host 2</td>
<td>S2</td>
</tr>
</tbody>
</table>

Here each table must contain the destination node address, the minimum number of hops to that destination and the next hop in the direction of that destination. The tables in DSDV also have an entry for sequence numbers for every destination. These sequence numbers form an important part of DSDV as they guarantee that the nodes can distinguish between stale and new routes. Here each node is associated with a sequence number and the value of the sequence number is incremented only by the node the sequence number is associated with. Thus, these increasing sequences numbers here emulate a logical clock. Suppose a node receives two updates from the same source, then the receiving node here makes a decision as to which update to incorporate in its routing table based on the sequence number. A higher sequence number denotes a more recent update sent out by the source node. Therefore it can update its routing table with more actual information and hence avoid route loops or false routes.

Having seen the table entries in DSDV, let us see how DSDV works. DSDV determines the topology information and the route information by exchanging these routing tables, which each node maintains. The nodes here exchange routing updates whenever a node detects a change in topology. When a node receives an update packet, it checks the sequence number in the
packet. If the information in the packet is older than the receiving node has in its routing tables, and then the packet is discarded. Otherwise, information is updated appropriately in the receiving node’s routing table. The update packet is then forwarded to all other neighboring nodes (except the one from which the packet came). In addition, the node also sends any new information that resulted from the merging of the information provided by the update packet. The updates sent out in this case, by nodes resulting from a change, can be of two types that is either a full update or a partial update. In case of full updates, the complete routing table is sent out and in case of a partial updates only the changes since last full update are sent out.

2.3.2.2 Dynamic Source Routing (DSR)

DSR uses a modified version of source routing. Operation of DSR can be divided in two functions – route discovery and route maintenance. Route discovery operation is used when routes to unknown hosts are required. Route maintenance operation is used to monitor correctness of established routes and to initiate route discovery if a route fails.

In DSR, when a node needs to send a packet to a destination it does not know about, the source node will initiate route discovery. The node sends route discovery request to its neighbors. The neighbors can either send a reply to the initiator or forward the route request message to their neighbors after having added their address to the request message (i.e. source routing). Every node that receives the route request message does the following:

- If the node has already seen this request, and then the request is discarded.
- If the node has not seen it, but the route request message already has address of this host, then also it is discarded.
Otherwise, if this host is the target of the route discovery message, then it appends the address of this host and returns it to the initiator of the route request message. The route request packet contains the route from the initiator to this host, which is the destination.

- If this host is not the destination, then just append the host’s address in the packet and forward it to all of hosts’ neighbors.

The route reply message can be returned to the initiator in two ways. If the host that sends reply already has the route to the initiator, it can use that route to send the reply. If not, it can use the route in the route request message to send the reply.

Route maintenance is performed when there is an error with an active route. When a node that is part of some route detects that it cannot send packets to next hop, it will create a Route Error message and send it to the initiator of data packets. The Route Error message contains the addresses of the node that sent the packet and of the next hop that is unreachable. When the Route Error message reaches the initiator, the initiator removes all routes from its route cache that have address of the node in error. It then initiates route discovery for a new route if needed.

2.3.2.3 Ad Hoc On-Demand Distance Vector (AODV)

Ad hoc On-Demand Distance Vector Routing is also an on demand routing algorithm, but in contrast to DSR it is a not source based routing scheme rather every hop of a route maintains the next hop information by its own. Operation of the protocol here is also divided in two functions route discovery and route maintenance. At first all the nodes send Hello message on its interface and receive Hello messages from its neighbors. This process
repeats periodically to determine neighbor connectivity. When a route is needed to some destination, the protocol starts route discovery. The source sends Route Request Message to its neighbors. If a neighbor has no information on the destination, it will send message to all of its neighbors and so on. Once request reaches a node that has information about the destination (either the destination itself or some node that has a valid route to the destination), that node sends Route Reply Message to the Route Request Message initiator. In the intermediate nodes (the nodes that forward Route Request Message), information about source and destination from Route Request Message is saved. Address of the neighbor that the Route Request Message came from is also saved. In this way, by the time Route Request Message reaches a node that has information to answer Route Request Message; a path has been recorded in the intermediate nodes. This path identifies the route that Route Request Message took and is called reverse path. Since each node forwards Route Request Message to all of its neighbors, more than one copy of the original Route Request Message can arrive at a node. When a Route Request Message is created at the initiator, it is assigned a unique id. When a node receives Route Request Message, it will check this id and the address of the initiator and discard the message if it had already processed that request.

A node that has information about route to the destination sends Route Reply Message to the neighbor from which it received Route Request Message. This neighbor then does the same. This is possible because of the reverse path created by the Route Request Message. While the Route Reply Message travels back using reverse path, that path is being transformed into forward path, by recording the node that Route Reply Message came from (i.e. same procedure as mentioned above just in opposite direction). When Route Reply Message reaches the initiator, the route is ready, and the initiator can start sending data packets.
If one of the links on the forward path breaks, the intermediate node just above the link that failed sends new Route Reply Message to all the sources that are using the forward path to inform them of the link failure. It does this by sending the message to all neighbors using the forward path. In turn, they will send to their neighbors until all upstream nodes that use forward path are informed. The source nodes can then initiate new route request procedures if they still need to route packets to the destination.

2.3.2.4 Temporally Ordered Routing Algorithm (TORA)

Temporally Ordered Routing Algorithm (TORA) is a distributed routing protocol based on a "link reversal" algorithm. It also discovers its routes on demand, provide multiple routes to a destination, establish routes quickly, and minimize communication overhead by localizing the reaction to topological changes when possible. Route optimality (shortest-path routing) is considered of secondary importance, and longer routes are often used to avoid the overhead of discovering newer routes. It is also not necessary (nor desirable) to maintain routes between every source/destination pair at all times.

2.3.2.5 Zone Routing Protocol (ZRP)

The Zone Routing Protocol (ZRP) is also called a hybrid ad hoc routing protocol as it combines proactive and reactive schemes. The ZRP maintains routing information for a local zone, and establishes routes on demand for destinations beyond its logical neighborhood. It limits the scope of the local zone by defining a maximum hop number for the local zone (e.g. 3 hops). Using ZRP with a maximum hop count of zero for the local neighborhood creates a reactive routing algorithm, and using it with maximum hop count as infinity creates a pure proactive routing algorithm. A route to a destination within the local zone can be established from the
proactively cached tables of the source node. The routing algorithm used in
the local zone can be based on any table driven routing algorithm, but it has to
be extended in a way, that packets contain the “time to live” (TTL)
information, which describes the maximum hop count of the local zone. For
routes beyond the local zone, route discovery happens reactively. The source
node sends a route request to its border nodes, containing its own address and
a unique sequence number. Border nodes are nodes, which are exactly the
maximum number of hops to the defined local zone away from the source.
The border nodes check their local zone for the destination. If the requested
node is not a member of this local zone, the node adds its own address to the
route request packet and forwards the packet to its border nodes. If the
destination is a member of the local zone of the node, it sends a route reply on
the reverse path back to the source. The source node uses the path saved in the
route reply packet to send data packets to the destination. The main advantage
of the ZRP is a reduced number of required RREQ messages and further on
the possibility to establish new routes without the necessity to completely
flood the network. The main disadvantage is the increased complexity of the
routing algorithm.

2.3.2.6 Cluster head Gateway Switch Routing Protocol (CGSR)

In CGSR, a set of nodes is divided into clusters as shown in
Figure 2.4. Each cluster has a node, which is designated as the cluster-head.
So, every node is either a cluster head or one wireless hop away from the
cluster head. A node that is not a cluster head, but adjacent to more than one
cluster head, is referred to as a gateway. Packets between cluster-heads are
routed through gateways. Finally, nodes that are neither cluster heads nor
gateways are referred to as ordinary nodes. The subnet comprising the cluster
heads and gateways is referred to as the backbone network. Here, each cluster
head maintains information about other nodes in its cluster, and from time to
time, this information is exchanged between cluster heads over the network.
Thus, the cluster heads gather network topology information. A node that has a packet to send to another node can obtain routing information from its cluster head. It is not necessary for a packet to be routed through the backbone network, as data packets may be routed along other more efficient routes in the network. In a dynamic network, cluster heads can cause performance degradation problems because of frequent cluster-head elections, so CGSR uses a Least Cluster Change (LCC) algorithm. In LCC, cluster head change occurs only if a change in network causes two cluster heads to come into one cluster, or if one of the nodes moves out of the range of all the cluster heads. When a source transmits the packet to its cluster head, it is forwarded to gateway nodes. And this continues until the destination is reached.

### 2.3.3 Recent Advances in Routing Algorithms

In the last section we explained some of the well known routing algorithms proposed for mobile ad hoc networks. In addition to these basic routing schemes, many variants of these routing algorithms have also been proposed. But in all these algorithms, the nodes were assumed to be blind; that is, there was no means by which nodes could get their spatial information with respect to other nodes in the network. In fact, the only way they could obtain information about the network topology was by sharing information among them. Recently, a different approach called the location-based (or position-based) routing was put forward. In addition to the topology-based information, these protocols also use information about the physical location of the mobile nodes, and to determine their respective positions, nodes often use the Global Positioning System (GPS). A recent survey, including comparative information on the time and communication complexities of various protocols of this category. The next stage of refinements in the routing algorithms came in when the existing traditional routing algorithms became inefficient in handling large ad hoc networks. Consider the average size of the routing table if the size of the network grew to 1000 nodes. Then
each node would be having a table containing 1000 records, which would measure up to considerable overhead when updating neighbors. And thus innovative routing schemes were needed to overcome such issues.

2.3.3.1 Location Aided Routing (LAR)

Location Aided Routing (LAR) is a variant of the DSR routing scheme and works on demand. The aim of the protocol is to limit route discovery flooding, so as to reduce control packet overhead. It accomplishes this by using GPS data. Using this information, the flooding is done within a defined rectangular region. Mobile nodes here can also store the speed of the target node, so that it can predict where the target node will possibly be at a later point of time.

A distinguishing characteristic of these location-based protocols is that, to forward packets, a node only requires three pieces of information, that is its own position, that of the destination, and those of its adjacent (one-hop) neighbors. A transmitting node uses a location service to determine the location of the destination, and includes this information as part of the destination address in its messages. Routes do not need to be established or maintained explicitly; thus, there is no need to store routing tables at the nodes, and no need for routing table updates. Typically adjacent nodes are identified by broadcasting limited range beaoning messages and by employing various time-stamping mechanisms. The beaoning message includes distance limits; a receiving node discards the message if its location lies beyond the distance limit.

The availability of accurate location information at each node is essential for location based routing to work, which, in turn, requires timely and reliable location updates as nodes change their locations. One or more nodes, designated to act as location servers, coordinate these location service
functions, which are necessarily decentralized because of mobility. A large part of the ongoing research is focused on designing efficient location services.

2.3.3.2 Global State Routing (GSR)

Global State Routing (GSR) is similar to DSDV, with changes to reduce the overhead, which normal DSDV would incur with increasing network sizes. In GSR, each node keeps only its neighbor's link state in its tables. The next hop table and distance table is also kept. The topology table contains the link state information as reported by the destination. The next hop table contains the next hop to which the packets for a destination node should be forwarded. The distance table keeps the shortest distance to each destination node.

As in all link state protocols, route update messages are generated upon link change. Upon receiving a routing message, a node updates its topology table if the sequence number of the message is higher than the value stored in the table. Fisheye methods, where the approach is to restrict the propagation of topology changes to a subset of the network if the effect is not so profound due to the change, works fine since it will work with both small and large networks. Thus with Fish eye method, nodes keep the most accurate topology information about the surrounding region, and know a little about outer regions. These methods are known to scale well.

2.3.3.3 Hierarchical State Routing (HSR)

Hierarchical State Routing (HSR) employs a multilevel clustering and logical partitioning scheme. The network is partitioned into clusters and a cluster-head is elected as in a cluster-based algorithm. Cluster heads again organize themselves into clusters up to any desired clustering level as shown
in Figure 2.5. Within a cluster, nodes broadcast their link information to one another. A cluster head summarizes its cluster information and sends it to neighboring clusters through a gateway node. A gateway node is one, which is adjacent to one or more cluster heads. Here cluster heads are members of a higher-level cluster. At each level, summarization and link information exchanges are performed. The way the information is exchanged in this hierarchy is, first information is collected among the nodes forming the base level cluster, it is then passed on to the cluster head which in turn passes to its next hierarchical cluster head and from there on the information is disseminated into other cluster heads and thus the information traverses down the hierarchy. Here each node has a hierarchical address, which may be obtained by assigning numbers from the top root to the bottom node. But as a gateway can be reached from the root from more than one path, so a gateway can have more than one hierarchical address.

Figure 2.5 An example of clustering, forming hierarchies
Also, each subnet contains a location management server (LMS). All nodes in the subnet are registered with the local LMS. LMS has to inform upper levels, and upper level information comes to local LMS server. When two nodes wish to communicate, they send their initial data to the LMS, and the LMS then forwards it to the destination. But if the source and destination know each other’s hierarchical addresses, they communicate directly. The protocol is highly adaptive to network changes. There are other clustering based studies that are adaptive to mobility; one such study relies on the probability of path availability to provide a basis for dynamically changing clusters.

2.3.3.4 Power and QoS Aware Protocols

As the acceptance of ad hoc networks grew, these networks were envisioned to support varied types of applications such as multimedia applications etc. Thus with growing applications the needs put forth in the routing algorithm became more and more constrained. Some applications needed QoS support, where as, some had security as their main concern and there were some which required the network to be up and running for a longer period of time and hence in such types of networks power efficiency became a major issue to be tackled. So in literature there are protocols that optimize parameters for power-efficient and these protocols consider the following metrics:

- Minimize energy consumed per packet: This implies the shortest-hop path.
- Maximize time to network partition: Here, the routing protocol tries to divide work amongst nodes to maximize life of the network.
• Maximize variance in node power levels: This implies the load-sharing of distributed systems, and is equivalent to bin-packing problem.

• Minimize cost per packet: Here, the goal is to maximize the life of individual nodes in the network, including costs other than energy.

• Minimize the maximum node cost: Here the minimization is done for per-node energy consumption of contributing nodes for forwarding packets.

There are also QoS related studies on ad-hoc networks. To provide QoS, there is a bandwidth reservation proposal that takes care of QoS parameters, such as to provide required bandwidth for real-time traffic on a route. Another study about QoS is Core Extraction Distributed Ad hoc Routing algorithm (CEDAR). In this protocol, QoS information is calculated at specific core nodes and they are propagated to other nodes forming the core of the network for overall QoS awareness in the network. A similar QoS related study, which introduce the QoS requirements of wireless mobile networks and make single modified routing protocol (e.g. DSDV) simulations, in general.

2.3.3.5 Comparison of the Routing Protocols

An introduction to ad hoc networks is presented and the general operating principles of the routing algorithms as well as the properties that are desirable in a routing protocol in ad hoc networks are discussed.
Table 2.2 Comparison of the ad-hoc routing protocols

<table>
<thead>
<tr>
<th>Protocol Property</th>
<th>DSDV</th>
<th>AODV</th>
<th>DSR</th>
<th>ZRP</th>
<th>TORA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loop Free</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Multiple Routes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Distributed</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Reactive</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Variable</td>
<td>Yes</td>
</tr>
<tr>
<td>Unidirectional Link Support</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>QoS Support</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Multicast</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Security</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Possible</td>
</tr>
<tr>
<td>Power Efficiency</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Periodic Broadcast</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Further in Table 2.2, the similarities and differences among some of the well-known routing protocols are presented.

2.4 AD HOC ROUTING PROTOCOLS ANALYSIS

The performance evaluation of existing Ad hoc routing protocols is done and the results are obtained from simulations, demonstrate that there is no single protocol suitable for all the Ad hoc networks. The performance of reactive, proactive and hybrid routing protocols in terms of scalability is highlighted.

2.4.1 Ad hoc Routing Protocols Evaluation

So far different routing protocols are described and based on the basic characteristics of reactive and proactive routing protocols a set of propositions are formulated. The propositions will consider the impact of
system variables such as used routing protocol type, node mobility and number of nodes (i.e. node density) on performance measures such as routing overhead, percentage of packet loss, end to end packet delay and percentage of optimal routes.

AODV, DSR and OLSR, are the experimental protocols standardized by the IETF as reactive and proactive routing protocols. The routing protocols under consideration in this evaluation are AODV and OLSR as the most representative of reactive and proactive categories.

2.4.2 Proactive versus Reactive Simulation Comparison

In this section, simulation results justifying the advantages and drawbacks of the reactive and proactive Ad hoc routing protocols are presented. The routing protocols comparison has been done using ns-2 simulator version 2.27 with standard IEEE 802.11 MAC protocol, which is used in the simulations.

The results are obtained from the average of three simulations rounds performed continuously in order to reduce any possible effect due to initialization process of the simulator.

In the simulations, the following parameters are considered:

- Simulation area: 1500m x 300m.
- Simulation time: 900 seconds.
- Traffic flows:
  - Constant Bit Rate (CBR) with UDP transport: 20 IP unidirectional flows.
  - Traffic with TCP transport: 20 IP unidirectional flows.
- Connection rate: 8 packets/second.
- Packet size: 65 bytes.
- Number of nodes: 50 nodes using random waypoint mobility pattern.
- Pause time between node movements: 0, 30, 60, 120, 300, 600 and 900 seconds.

In the simulations, the mobility as the average speed of the node during the simulation is considered. Simulations are run with the same parameters but using either UDP or TCP as transport protocol for the traffic flows to compare the effect of congestion and reliable traffic control mechanisms.

Ad hoc networks will be deployed under different mobility patterns and the routing protocols have to perform in different environments. Therefore, in the simulations, the nodes follow a different mobility pattern after each waiting time as characterized in the random waypoint model. The simulations are made considering that the network is handling the traffic generated by 20 active connections transmitting 8 packets/second.

The simulations reflect the performance of Ad hoc networks with real time applications under different mobility conditions and using different routing and transport protocols.

The simulations last for 900 seconds, thus a pause time of 900 seconds is equivalent to static nodes that do not move during the simulation.

Both reactive (i.e. AODV, TORA, DSR) and proactive routing protocols (i.e. DSDV, OLSR) are covered in the simulations. The simulation results presented in this section are inaccurate due to the random behaviour of the nodes. Therefore, a deeper analysis will be made extracting from each
simulation the associated equation for the most representative reactive (i.e. AODV) and proactive (i.e. OLSR) routing protocols and specific transport protocol (i.e. TCP or UDP).

Figure 2.6 shows the routing overhead generated by reactive and proactive routing protocols during the simulation time versus node mobility with UDP traffic flows and tabulated in Table 2.3.

**Table 2.3 Routing overhead versus node mobility**

<table>
<thead>
<tr>
<th>Pause time in sec</th>
<th>Routing overhead in Kbytes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TORA</td>
</tr>
<tr>
<td>900</td>
<td>190</td>
</tr>
<tr>
<td>600</td>
<td>190</td>
</tr>
<tr>
<td>300</td>
<td>190</td>
</tr>
<tr>
<td>120</td>
<td>180</td>
</tr>
<tr>
<td>60</td>
<td>200</td>
</tr>
<tr>
<td>30</td>
<td>210</td>
</tr>
<tr>
<td>0</td>
<td>220</td>
</tr>
</tbody>
</table>

**Figure 2.6 Routing overhead versus node mobility**
Proactive protocols have a higher routing overhead than reactive protocols, which can be caused by the additional topology information they exchange. In particular, AODV generates less routing overhead compared to OLSR in similar conditions.

Figure 2.7 shows the routing overhead in AODV and OLSR using a transport protocol that includes reliability and congestion mechanisms such as TCP and the readings are tabulated in Table 2.4. The routing overhead increases in both AODV and OLSR compared to UDP traffic flows.

Figure 2.8 shows the end-to-end packet delay generated by reactive and proactive routing protocols during the simulation time versus node mobility with UDP traffic flows and the readings are tabulated in Table 2.5. In high mobility conditions, proactive routing protocols such as OLSR present higher delay than reactive routing protocols. In case of low mobility, performance of reactive and proactive routing protocols is similar.

### Table 2.4 Routing overhead versus node mobility and transport protocol

<table>
<thead>
<tr>
<th>Pause time in sec</th>
<th>Routing overhead in Kbytes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AODV(UDP)</td>
</tr>
<tr>
<td>900</td>
<td>130</td>
</tr>
<tr>
<td>600</td>
<td>130</td>
</tr>
<tr>
<td>300</td>
<td>130</td>
</tr>
<tr>
<td>120</td>
<td>130</td>
</tr>
<tr>
<td>60</td>
<td>130</td>
</tr>
<tr>
<td>30</td>
<td>130</td>
</tr>
<tr>
<td>0</td>
<td>130</td>
</tr>
</tbody>
</table>
Figure 2.7 Routing overhead versus node mobility and transport protocol

Table 2.5 End to end packet delay versus node mobility

<table>
<thead>
<tr>
<th>Pause time in sec</th>
<th>TORA</th>
<th>DSR</th>
<th>AODV</th>
<th>OLSR</th>
<th>DSDV</th>
</tr>
</thead>
<tbody>
<tr>
<td>900</td>
<td>0.033</td>
<td>0.012</td>
<td>0.03</td>
<td>0.00</td>
<td>0.018</td>
</tr>
<tr>
<td>600</td>
<td>0.012</td>
<td>0.027</td>
<td>0.045</td>
<td>0.03</td>
<td>0.045</td>
</tr>
<tr>
<td>300</td>
<td>0.045</td>
<td>0.10</td>
<td>0.039</td>
<td>0.075</td>
<td>0.066</td>
</tr>
<tr>
<td>120</td>
<td>0.012</td>
<td>0.23</td>
<td>0.066</td>
<td>0.233</td>
<td>0.075</td>
</tr>
<tr>
<td>60</td>
<td>0.145</td>
<td>0.19</td>
<td>0.033</td>
<td>0.80</td>
<td>0.239</td>
</tr>
<tr>
<td>30</td>
<td>0.109</td>
<td>0.148</td>
<td>0.075</td>
<td>0.60</td>
<td>0.233</td>
</tr>
<tr>
<td>0</td>
<td>0.015</td>
<td>1.05</td>
<td>0.084</td>
<td>0.97</td>
<td>0.354</td>
</tr>
</tbody>
</table>
Figure 2.8 End to end packet delay versus node mobility

Node mobility affects the end-to-end packet delay because of different reasons such as network congestion and loss of connectivity. Network congestion increases with mobility due to the link breaks that generate new topology updates in proactive protocols, and additional route requests initiated in reactive protocols. The connectivity is immediately re-established after the link break by reactive protocols but the same is performed after a periodic route update in proactive protocols.

Figure 2.9 shows that the end-to-end delay is reduced using TCP as transport protocol and the values are recorded in Table 2.6. This can be due to the fact that with TCP both ends maintain a connection state, thus they will notice a link break immediately and either trigger a route update earlier than the normal periodic update, or they will recalculate an alternative route in the routing table. The difference in reactive protocols when using either UDP or TCP is minor because reactive protocols do not maintain routing tables. They do not have alternative routes available to re-route the traffic and they just issue a route request when needed. The reactive protocols have similar
behavior with UDP and TCP because they detect the link break immediately and initiate the route discovery to provide an alternative path.

Figure 2.10 shows the percentage of packet loss generated when reactive or proactive routing protocols are used during the simulation time versus node mobility with UDP traffic flows and the readings are tabulated in Table 2.7.

The packet loss is measured as the percentage of packets that did not reach the destination from the total number of packets sent. The percentage of packet loss is higher in case of proactive routing protocols than in case of reactive routing protocols and increases with mobility.

Table 2.6 End to end packet delay versus node mobility and transport protocol

<table>
<thead>
<tr>
<th>Pause time between movements (seconds)</th>
<th>End to End packet delay</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AODV(UDP)</td>
</tr>
<tr>
<td>900</td>
<td>0.03</td>
</tr>
<tr>
<td>600</td>
<td>0.05</td>
</tr>
<tr>
<td>300</td>
<td>0.06</td>
</tr>
<tr>
<td>120</td>
<td>0.08</td>
</tr>
<tr>
<td>60</td>
<td>0.04</td>
</tr>
<tr>
<td>30</td>
<td>0.07</td>
</tr>
<tr>
<td>0</td>
<td>0.08</td>
</tr>
</tbody>
</table>
Figure 2.9  End to end packet delay versus node mobility and transport protocol

Table 2.7 Percentage of packet loss versus node mobility

<table>
<thead>
<tr>
<th>Pause time between movements (seconds)</th>
<th>Packet loss %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DSDV</td>
</tr>
<tr>
<td>900</td>
<td>4</td>
</tr>
<tr>
<td>600</td>
<td>11</td>
</tr>
<tr>
<td>300</td>
<td>16</td>
</tr>
<tr>
<td>120</td>
<td>24.5</td>
</tr>
<tr>
<td>60</td>
<td>30</td>
</tr>
<tr>
<td>30</td>
<td>26</td>
</tr>
<tr>
<td>0</td>
<td>29.5</td>
</tr>
</tbody>
</table>
2.4.3 Ad hoc Routing Protocols Simulation Conclusions

The reactive routing protocols under analysis have clear drawbacks such as the excessive flooding traffic in the route discovery and the route acquisition delay. When the network is congested, the routing information is lost and a consecutive set of control packets are issued to re-establish the links, increasing the routing latency (i.e. time the routing protocol requires for obtaining the route to the destination node) and percentage of packet loss. If the Hello messages are not received, then error requests are issued and new route requests are sent to re-establish the link. Thus, the reactive protocols do not scale when the load and node density increase. Moreover, the reactive routing protocols do not have knowledge about the QoS in the path before the route is established and the routes are not optimized.

The reactive routing protocols suffer from high routing latency and percentage of packet loss, which increase with mobility and large networks. The percentage of optimal routes calculated with reactive protocols is lower.
than in proactive protocols and it decreases in large networks. An advantage of reactive protocols like AODV is that they maintain only the active routes in the routing table, which minimizes the memory required in the node. Moreover, the protocol itself is simple so the computational requirements are minimal, extending the lifetime of the node in the Ad hoc network. The routing overhead is equivalent to additional packet processing, thus reactive protocols will have lower power consumption than proactive protocols. In simulations with a small number of nodes, AODV has lower percentage of packet loss than OLSR. Therefore in networks with light traffic and low mobility reactive protocols are scalable because of the small bandwidth and storage requirements.

The proactive routing protocols under analysis maintain topology information up to date with periodic update messages. The proactive routing protocols minimize the route discovery delay, which minimizes the percentage of packet loss since the routes are known in advance and no additional routing overhead and processing are required. However, under high mobility conditions more and more routes established based on the previous periodic update become stale leading to an increased percentage of packet loss.

The proactive routing protocols have low routing latency since all the routes are available immediately even in large networks. The proactive routing protocols calculate the most optimal routes since they apply hop count based routing algorithms. The proactive routing protocols have higher percentage of packet loss than reactive protocols in networks with reduced number of nodes and high mobility as depicted in Figure 2.10.

A drawback of proactive routing protocols is that they require a constant bandwidth and cause a processing overhead to maintain the routing information up to date. This overhead increases with the number of nodes and
mobility since the updates have to be more frequent to maintain accurate routing information. The proactive routing protocols have lower routing latency but they do not react quickly enough to topology changes. The proactive routing protocols have been enhanced towards hybrid and hierarchical solutions to deal with this scalability problem in Ad hoc networks. OLSR reduces the control and processing overhead by selecting some nodes (i.e. Multipoint Relay nodes) within the network to maintain the routing information. The link information updates are propagated between MPR (Multipoint Relay) nodes only, reliving the rest of the nodes from participating in the topology maintenance. Other optimizations consist of exchanging only the differential updates, implementing hybrid solutions such as ZRP that combines reactive and proactive routing protocols or routing protocols that use the nodes location data such as LAR.