Chapter 3

Background and Related Work

Objectives of the Chapter

- To introduce Mobile Ad hoc Network
- Routing protocols used in a mobile ad hoc network
- Mobility modelling in a mobile ad hoc network
- Clustering in a mobile ad hoc network
- Works related to the modelling of the ad hoc network perimeter

3.1 Introduction to Mobile Ad Hoc Networks

A Mobile Ad hoc Network (MANET) [77] is an autonomous system of mobile routers as well as hosts connected by wireless links which form an arbitrary graph when taken together. The mobile hosts are free to move randomly and organize themselves arbitrarily. The network’s wireless topology may change rapidly from time to time. Such kind of network may operate in a stand alone fashion, or may be connected to the larger network such as the Internet. Ad hoc networks are characteristically different because they do not require any prior investment in fixed infrastructure. Instead, the network nodes are assumed to agree
3.1 Introduction to Mobile Ad Hoc Networks

Figure 3.1 A Typical mobile ad hoc network topology

to relay each other’s packet toward their ultimate destinations, and the nodes automatically form their own cooperative infrastructure. An example of a typical MANET topology is shown in Figure 3.1. MANET requires very little efforts to deploy, making it very attractive for short term use like rescue work, conference etc. [77]. These networks are indispensable for emergency services such as disaster recovery where wireless access to a wired backbone is either ineffective or unavailable. These networks are a valid substitution for local area networks as well. Nodes in an ad hoc mobile network forward packets to establish a virtual network backbone. The idea of forwarding each other’s packet eliminates the need for a fixed network for communication. Few configuration requirements are also a point of attraction for ad hoc mobile network, which makes it suitable for home networks or users who either do not know how to configure a network or do not have a desire to do so.

Though ad hoc networks are attractive, they also have their own disadvantages compared to fixed networks. Fixed networks take advantage of their static architecture in two ways. First, they proactively distribute network topology information among the nodes, and each node is able to precompute routes through that topology using relatively inexpensive and less complex algorithms. Second, fixed networks embed routing hints in node addresses because the complete topology of a large network is too unwieldy to process or distribute globally. Neither of these techniques works well for ad hoc networks with mobile nodes.
because movement invalidates topology information and permanent node addresses cannot
include dynamic location information. The idea of packet forwarding, although works well
in the case of small networks, however with increasing network size each node has to divert
a significant amount of computing power for forwarding other node’s packets. In the future
these networks will need to support multimedia traffic such as voice, video and data, like
3G network and Wi-Fi. To fulfil these requirements, networks carrying data in real time
and having high throughput, low delay and fault tolerance are needed which can not be met
using existing methods. Another problem with ad hoc mobile networks is their inability to
deal with the node density. If the network becomes crowded, bandwidth share of each node
will decrease and workload on each of them will increase due to the increasing number of
packets to be forwarded and location updates subsequently. If the node density in a specific
area drops too low the probability of a message encountering dead-ends in the network
increases. The whole network may be divided into different sub-networks which can not be
connected to each other. The following are the characteristics that are identified in mobile
ad hoc networks:

1. New members join and existing members may leave the network any time.

2. No base station to provide connectivity to backbone hosts or to other mobile hosts.

3. No need for handover and location management.

4. Each mobile host is capable of behaving as a router, forwarding packets from one
mobile host to another.

5. Communication connectivity is fairly weak.

### 3.2 Routing Protocols

Developing support for routing is one of the most significant challenges in ad hoc networks
and is critical to the basic network operations. Routing in MANET is challenging because
of the dynamically changing topology of the network, lower bandwidth, energy efficiency
etc. First, nodes in an ad hoc network are allowed to move in an uncontrolled manner.
Such node mobility results in a highly dynamic network with rapid topological changes
causing frequent route failures. A good routing protocol for this network environment has to dynamically adapt to the changing network topology.

Second, the underlying wireless channel provides much lower and more variable bandwidth than wired networks. The wireless channel working as a shared medium makes available bandwidth per node even lower. So routing protocols should be bandwidth-efficient by expending a minimal overhead for computing routes so that much of the remaining bandwidth is available for the actual data communication.

Third, nodes run on batteries which have a limited energy supply. In order for nodes to stay and communicate for longer periods, it is desirable that a routing protocol be energy-efficient as well. This also provides another reason why overheads must be kept low. Thus, routing protocols must meet the conflicting goals of dynamic adaptation and low overhead to deliver good overall performance. Routing protocols developed for wired networks such as the wired Internet are inadequate here as they not only assume mostly fixed topology but also have high overheads. This has lead to several routing proposals specifically targeted for ad hoc networks \[49, 72, 74, 75, 79, 94\]. Researchers traditionally classify these protocols as proactive protocols, reactive protocols and hybrid of the two, based on the way they find new routes or update existing ones. Proactive routing protocols keep routes updated continuously, while reactive routing protocols react to demand. Routing protocols can also be classified as link state protocols or distance vector protocols. Routers using a link state routing protocol maintain a full or partial copy, of the network topology and costs for all known links. Routers using a distance vector protocol keep only information about next hops to the adjacent neighbours and costs for paths to all known destinations. We briefly discuss three popular routing protocols viz. (i) Ad Hoc On-Demand Distance Vector \[74\], (ii) Destination sequenced distance Vector \[75\] and (iii) Dynamic source routing \[47\] as we have used these protocols only in our present scope.

### 3.2.1 Ad hoc On-Demand Distance Vector (AODV)

AODV is an effective example of reactive or on-demand routing protocol. It uses an on-demand approach for finding routes i.e. a route is established only when it is requested by a source node for transmitting data packets to the receivers and these routes are maintained until they are in need by the source. In AODV, each node maintains at most one route per
destination. Being a single path protocol, it has to invoke a new route discovery, whenever the only path from the source to the destination fails. When topology changes frequently, route discovery needs to be initiated again and again, which can be very inefficient. AODV maintains a destination sequence number generated by the receivers and determines an up-to-date path to the destination. A node updates its route information only if the destination sequence number of the current packets received is greater than the destination sequence number stored at the node. It indicates the freshness of the route accepted by the source. To prevent multiple broadcast of the same packet, AODV uses a broadcast identifier number that ensures loop freedom. This is because the intermediate nodes only forward the first copy of the same packet and discard the duplicate copies. When a route to a new destination is needed, the node uses a broadcast route request (RREQ) to find a route to the destination. Nodes on receipt of the RREQ, find out whether they are the destination or whether they have a fresh route to the destination. Then they respond to the RREQ by unicasting a route reply (RREP) back to the source node. A route can be determined when the request reaches either the destination itself, or an intermediate node with a fresh enough route to the destination. Since each node receiving the request keeps track of a route back to the source of the request, the RREP can be unicast back from the destination to the source, or from any intermediate node that is able to satisfy the request back to the source. Figure 3.2 shows the route establishment of AODV routing protocol. In Figure 3.2, we have shown that the path is 1-4-6. This is because the route reply is 6-4-1 from destination to source. AODV has a concept of route cache. If some intermediate node holds the path from that node to the destination, then AODV may establish a path from the source node to that route cache.
If we consider a route node 2 holds cache route (5-6) then 1-2-5-6 may be another path.

### 3.2.2 Dynamic source routing (DSR)

Dynamic source routing (DSR) is another example of a reactive routing protocol. It generates the proper route only when a packet needs to be forwarded from source to destination. Within the limit of the transmission range, the process of finding a path is only executed when a path is needed by a node. DSR makes aggressive use of source routing and route caching. With source routing, complete path information is available and routing loops can be easily detected and eliminated without requiring any special mechanism. Because route requests and replies are both source routed, the source and destination, in addition to learning routes to each other, can also learn and cache routes to all intermediate nodes. The addresses of intermediate nodes in the route are kept within the delivered packets. The route discovery process broadcasts a ROUTE REQUEST packet that is flooded across the network in a controlled manner. ROUTE REQUEST packets use sequence numbers to prevent duplication. The request is answered by a ROUTE REPLY packet either from the destination node or an intermediate node that has a cached route to the destination. To take full advantage of route caching, DSR replies to all requests reaching a destination from a single request cycle. Thus, the source learns many alternate routes out to the destination, which will be useful in case the primary route fails. Figure 3.3 shows the route establishment in DSR protocol. DSR is a source routing protocol. It maintains multiple routes to a destination and uses immoral mode listening. The current specification for DSR does not contain
any mechanism for route entry invalidation. This leads to stale cache entries, predominantly at high mobility. In Figure 3.3, we have shown the route establishment in DSR routing protocols. The route reply paths are 6-4-1, 6-5-2-1 and 6-3-1 from destination to source. So, the established routes are 1-4-6, 1-2-5-6 and 1-3-6.

### 3.2.3 Destination Sequenced Distance Vector (DSDV)

One of the first ad hoc routing protocols was Destination Sequenced Distance Vector (DSDV) routing protocol. In DSDV, each node has a routing table which contains the details of the next hop for each destination it can reach, along with the number of hops to that destination and a sequence number. The sequence numbers are used to distinguish stale routes from new ones, thus avoiding the formation of routing loops. When a destination node sends out a broadcast, it assigns it a unique sequence number higher than the last, thereby ensuring that receivers know this is new information from the destination. Any nodes receiving an update, will use the highest sequence number. If two updates have the same sequence number, the route with the smallest cost metric will be used. DSDV alleviates table update flooding by employing two types of update packets. A "full dump" packet carries all available routing information. It is used when (i) network changes are high, (ii) consistency checking is required, and (iii) new nodes join the network. Smaller "incremental" packets are used to relay information that has been undergoing changes since the last "full dump". Aside from the sequence numbers, DSDV follows the common table design of generic distance vector routing protocols. Implementing two different broadcasts of routing information, "full dump" and "incremental", cuts down the amount of traffic generated. Lesser traffic reduces the packet collisions and hence saves the battery power which is vital in MANET. DSDV successfully introduces distance vector routing to the wireless ad hoc environment.

### 3.3 Mobility Models

Node mobility is a prominent feature of ad hoc networks. As a consequence, studying the performance of ad hoc networking protocols in the presence of mobility is a fundamental stage of the design process. Since real implementations of ad hoc networks are scarce, real-life movement patterns are very difficult to obtain, and the common approach is to use
synthetic mobility models and simulation.

A mobility model is a set of rules used to generate trajectories for mobile entities. Mobility models are used in network simulations to generate network topology changes due to movement of nodes in the environment. Mobility models for ad hoc networks should

**resemble real-life movements:** Given the wide range of ad hoc network applications, the movement patterns to consider are numerous: they range from campus wide movement of students into vehicular motion in highways, from movement of groups of tourists in an urban scenario to rescue squads motion in disaster areas, and from sensors carried around by ocean flows to animal movement in animal tracking wireless sensor network (WSN) applications. Providing a unique mobility model that resembles all these types of mobility is virtually impossible. However, a mobility model should be representative of at least one application scenario.

**simple enough for simulation/analysis:** Since mobility models are used in the simulation of ad hoc networks, the model should be simple enough to be integrated into the simulator and to keep the simulation running time reasonable. Furthermore, using relatively simple mobility models eases the task of deriving meaningful analytical results concerning fundamental network parameters in presence of mobility. In turn, these results can be used to optimize the performance of ad hoc networking protocols.

Clearly, the two goals above are conflicting: the more realistic the model is, the more the details that must be included in it, and the model complexity increases. Thus, a synthetic mobility model should be a good compromise between representativeness and simplicity, that is, it should consider the salient features of a certain movement pattern, while disregarding secondary details. In this section, we briefly describe the most important mobility models such as Random Way Point, Reference Point Group, Manhattan Grid, Gauss Markov etc. used in the simulation of ad hoc networks.

### 3.3.1 Random walk mobility

Random walk mobility models are used in various scientific disciplines such as the path traced by a molecule as it travels through a liquid, the trajectory of a foraging animal, the
financial status of a gambler, the time-varying price of a stock or share, and so on. The term random walk was first introduced by the mathematician Karl Pearson in 1905 [73]. It refers to a movement trajectory consisting of successive random steps. Random walk is further described in [11, 65] for MANET. It defines that a mobile node moves from its current location to a new location by randomly choosing a direction and speed for travelling. The new speed and direction are both chosen from pre-defined ranges, \([\text{speedmin}; \text{speedmax}]\) and \([0; 2\pi]\) respectively, where \text{speedmin} is the minimum speed allowed to the nodes and \text{speedmax} is the maximum allowable speed for the nodes. Each move in the Random Walk Mobility Model occurs in either a constant time interval \(t\) or a constant distance travelled \(d\). At the end of a move, a new direction and speed are calculated. If a node which moves according to this model reaches a simulation boundary, it bounces off the simulation border with an angle determined by the incoming direction. There exist many variations of the random walk mobility model. For instance the Random Direction Mobility Model forces nodes to travel to a boundary of the simulation area before changing direction and speed.

### 3.3.2 Gauss Markov Mobility Model (GM)

The Gauss-Markov Mobility Model was originally proposed for the simulation of a personal communication service (PCS) networks [57]. The Gauss-Markov Mobility Model was designed to adapt to different levels of randomness via one tuning parameter \(\alpha\). Initially, each mobile node is assigned a current speed and direction. At fixed intervals of time, \(n\), movement occurs by updating the speed and direction of each mobile node. Specifically, the value of speed and direction at the \(n^{th}\) instance is calculated based upon the value of speed and direction at the \((n - 1)^{st}\) instance and a random variable using the following equations:

\[
\begin{align*}
    s_n &= \alpha s_{n-1} + (1 - \alpha)\bar{s} + \sqrt{(1 - \alpha^2)s_{x n-1}} \\
    d_n &= \alpha d_{n-1} + (1 - \alpha)\bar{d} + \sqrt{(1 - \alpha^2)d_{x n-1}}
\end{align*}
\]  

(3.1) (3.2)

where \(s_n\) and \(d_n\) are the new speed and direction of the mobile node at time interval \(n\); \(\alpha\), is the tuning parameter used induce the randomness; \(\bar{s}\) and \(\bar{d}\) are constants representing the mean value of speed and direction of mobile nodes; and \(s_{x n-1}\) and \(d_{x n-1}\) are random
variables from a Gaussian distribution. Totally random values (or Brownian motion) are obtained by setting $\alpha = 0$ and linear motion is obtained by setting $\alpha = 1$. Intermediate levels of randomness are obtained by varying the value of $\alpha$ between 0 and 1. At each time interval the next location is calculated based on the current location, speed, and direction of movement. Specifically, at time interval $n$, a mobile node’s position is governed by the equations:

$$x_n = x_{n-1} + s_{n-1} \cos(d_{n-1})$$ (3.3)

$$y_n = y_{n-1} + s_{n-1} \sin(d_{n-1})$$ (3.4)

where $(x_n;y_n)$ and $(x_{n-1};y_{n-1})$ are the $x$ and $y$ coordinates of the mobile nodes position at the $n^{th}$ and $(n-1)^{st}$ time intervals, respectively, and $s_{n-1}$ and $d_{n-1}$ are the speed and direction of the mobile node, respectively, at the $(n-1)^{st}$ time interval. To ensure that a mobile node does not remain near an edge of the grid for a long period of time, the mobile nodes are forced away from an edge when they move within a certain distance of the edge. This is done by modifying the mean direction variable $\bar{d}$ in the above direction equation. For example, when a mobile node is near the right edge of the simulation grid, the direction $\bar{d}$ is changed to 180 degrees. Thus, the mobile node finds a new direction away from the right edge of the simulation grid.

$\alpha$ plays an important role in defining this mobility model. If $\alpha = 0$, we have speed and direction values at a time step $t + 1$ which are completely uncorrelated with those in time $t$, and the model reduces to a form of Brownian motion. On the other hand, when $\alpha = 1$ we have maximum correlation between the speed and direction values at times $t$ and $t + 1$, since $v(t + 1) = v(t)$ and $d(t + 1) = d(t)$. Intermediate values of parameter $\alpha$ correspond to intermediate degrees of temporal correlation in the speed and direction stochastic processes.

### 3.3.3 Manhattan Grid mobility model (MHG)

The Manhattan Grid mobility model was introduced in [3] by Bai et. al.. The Manhattan Grid mobility model uses a grid road topology. The network is assumed to be divided into several grids where all the square blocks in the grid are equal to the block length. The nodes are placed on the different roads randomly at the starting of the simulation.
moving direction of the nodes is decided by the initial location in the road and the starting position of the node. The nodes position and the direction of selected movement determine the distance to the next intersection or crossover of the road. When a node reaches one of the intersections of the road, then its position is corrected to remove round-off errors and the new direction of movement is randomly chosen. In general, the node leaves the scenario and re-enters at a randomly chosen new road point. However, each outgoing node is replaced by a newly arrived node, which enters the scenario at a randomly chosen crossover point. The node density of the mobility scenario remains same even if nodes are replaced constantly. In this mobility model, the mobile nodes are placed anywhere along the street and the movement of a node is decided one step horizontally or vertically at a time. Initially, each node may choose any of the streets. The center nodes may move in four possible directions like north, south, east, and west. This means a node would have a 25% chance of moving in one of the directions say north, leaving another node for moving towards west, east or south. The velocity of a mobile node at a time slot is totally dependent on its velocity at the previous instant of time. In a Manhattan mobility model, the velocity of a node is always limited by the velocity of the node preceding it on the same lane of the street. This is because nodes have to maintain a safe distance on the same lane to avoid any collisions.

The entry point of the nodes in the simulation region is called ingress point and the exit point of the nodes from the simulation region is called egress point as depicted in Figure 3.4 in the Manhattan mobility model. Nodes enter the simulation region from a randomly chosen ingress point, then they move along the selected lane until they exit the simulation region from the corresponding egress point.

### 3.3.4 Random Way Point mobility model (RWP)

Random way point is the most popular mobility model in mobile ad hoc network simulation. In the random waypoint mobility model, each node randomly chooses a destination location in the simulation area and moves towards this destination with a randomly chosen velocity. When the destination is reached, the station remains at the same place for a while. Once this time span expires, the node chooses a random destination in the simulation area and a speed that is uniformly distributed in $[\text{minspeed, maxspeed}]$ where $\text{speedmin}$ is the
minimum speed allowed to the nodes and $speed_{max}$ is the maximum allowable speed for the nodes. The node then travels toward the newly chosen destination at the selected speed. This process is repeated by each station until the end of the simulation. Often in the model, the nodes are initially distributed randomly around the simulation area. As long as the simulation runs, the space distribution of the nodes tends to some stable value, referred to as its stationary distribution. It is the default mobility pattern incorporated in NS-2 simulator.

An example of RWP mobility in the unit square $R = [0, 1]^2$ is shown in Figure 3.5. The starting point in the referred Figure 3.5 is the black dot. The next waypoint is chosen uniformly at random in $R$ with the node moving along a linear trajectory. The trajectory followed by the node during 20 mobility steps is illustrated in Figure 3.5.

It has been recently discovered that the long-term node spatial distribution of RWP mobile networks is concentrated in the center of the deployment region. This is termed as the border effect [10, 12, 13]. The average nodal speed, defined as the average of the node velocities at a given instant of time, decreases over time [101]. These observations have brought to the attention of the community the fact that RWP mobile networks must be carefully simulated. In particular, network performance should be evaluated only after a certain warm-up period, which must be long enough for the network to reach the node spatial and average velocity steady-state distribution.
3.3 Mobility Models

3.3.5 Random direction model

The random direction model resembles individual, obstacle-free movement similar to RWP mobility. This model was created to maintain a uniform node spatial distribution during the simulation time, thus avoiding the border effect typical of RWP mobility. In this model [76], any node chooses a direction uniformly at random in the interval $[0, 2\pi]$, and a random velocity in the interval $[v_{\text{min}}, v_{\text{max}}]$. Then, it starts moving in the selected direction with the chosen velocity. When the node reaches the boundary of $R$, it chooses a new direction and velocity, and so on.

3.3.6 Reference Point Group Mobility Model(RPGM)

The real applications where Reference Point Group Mobility model(RPGM) model can be used properly is the mobility behaviour of the soldiers moving together in a group in the battle field. There is a logical center or group leader in each group. The movement of the group leader determines the mobility behaviour of all other members in the group. Initially, each member of the group is uniformly distributed in the neighbourhood of the group leader. At every time instant, each node has its own speed and direction which is deviated randomly from that of the group leader. This model realizes the spatial dependency of each node of a group with logical center. The movement of the group leader at time $t$ can be represented
3.4 Clustering

Figure 3.6 An example of RPGM mobility for a group of three nodes.

by motion $V_{\text{group}}^t$. Not only does it define the motion of the group leader itself, but also provides the general motion trend of the whole group under its control. Each member of this group deviates from this general motion $V_{\text{group}}^t$ by some degree. The motion vector $V_{\text{group}}^t$ can be randomly chosen or carefully designed based on certain predefined paths. The movement of group members is significantly affected by the movement of the group leader. For each node, mobility is assigned to a reference point that follows the group movement. Based on this predefined reference point, each mobile node could be randomly placed in its neighbourhood. Formally, the motion vector $V_i^t$ of group member $i$ at time $t$, can be described as,

$$V_i^t = V_{\text{group}}^t + R \times M_i^t$$  \hspace{1cm} (3.5)

where the motion vector $R \times M_i^t$ is a random vector deviated by group member $i$ from its own reference point at time $t$. An example of RPGM mobility for a group composed of three nodes is shown in Figure 3.6.

3.4 Clustering

Clustering in MANET can be defined as the virtual partitioning of the dynamic nodes into various groups. Groups of the nodes are made with respect to their nearness to other nodes. Two nodes are said to be neighbour of each other when both of them lie within their respective transmission range and set up a bidirectional link between them. The cluster control structure forms the virtual backbone of communication where cluster heads are the communication hot spots. The cluster head works as the local coordinator for its member nodes.
and does the resource management among them similar to a base station of cellular architecture. These cluster heads are responsible for inter cluster and intra-cluster communication. Inter-cluster communication is made possible through the gateway nodes. A Cluster gateway node is a node that works as the common or distributed access point for two cluster heads. When a node lies within the transmission range of two cluster heads and supports inter-cluster communication, it is called the ordinary gateway for two corresponding clusters. A typical case of cluster structure in MANET is shown in Figure 3.7. The nodes labelled CH denotes cluster heads. The nodes with label GW are gateway Node. The white nodes are member nodes attached to the cluster head of that circle shown in Figure 3.7. Depending on the diameter of the clusters, there exist two kinds of cluster control architectures, known as one-hop clusters and multi-hop \((d\)-hop\) clusters. In one-hop clusters, every member node is at most 1-hop distance away from the central coordinator or the cluster head. Thus, all the member nodes remain at most two hops distance away from each other within a logical cluster. But in multi hop clusters, the constraint of the immediate neighbourhood of members from the head is eliminated by allowing the nodes to be present at most \(d\)-hop distance away from each other to form a cluster [58, 59, 70].

The process of clustering can be visualized as a combination of two phases, i.e., cluster formation and cluster maintenance. The cluster formation phase deals with the logical partition of the mobile nodes into several groups and selection of a set of suitable nodes to act as heads in every group. In a mobile ad hoc network, where the topology changes frequently, selection of optimum number of cluster heads is an NP-hard problem. There exist
some representative algorithms that use the parameters like node identity number, mobility, battery power, degree of connectivity etc. as the factors to decide its suitability for becoming cluster head [8]. Even some researchers combine multiple node parameters to select this set of routers in an efficient manner. These selected nodes are responsible for routing as well as node management in the mobile network and collectively called as the dominant set in graph theory terminology.

The objective of cluster maintenance is to preserve the existing clustering structure as much as possible. In one hop clustering, since every node is directly connected to a cluster head, the mobility of either the member node or the cluster head may drive them away from each other. There exists a bidirectional link between these two nodes till both of them are within their transmission range. When any of them moves away from the other, a link failure occurs and the member node searches for another new head within its transmission range to get affiliated to. This kind of situation is called as re-affiliation with a new head node.

The requirement for the re-election of cluster heads arises when the current heads fail to cover all the nodes in the network. Sometimes a node may move away from the transmission range of all the current cluster heads and becomes an orphan node. This demands a re-election of cluster heads. Even at times any of the cluster heads may drain out of energy or may even fail to work due to any fault occurrence and needs a head re-election process. However, such an unavoidable re-election increases the computational cost and the message transmission overhead.

Basagni et. al. presented a Distributed Clustering Algorithm (DCA) [7] that is mobility adaptive and truly distributed in nature. This algorithm is a generic weight based cluster formation algorithm. Here each node is associated with a parameter called the weight (i.e. $\geq 0$) that decides the role of a node. The weight of a node may be the function of its transmission range or node mobility. DCA does not allow the change in network topology during the execution of the algorithm. A node having bigger weight among all its one-hop neighbours is selected as the cluster head (ties are broken by using Lowest ID). An ordinary node opts to join a cluster head with the biggest weight when it comes across several other heads in its proximity. This algorithm explains well for the cluster formation, whereas the maintenance of the clusters in the presence of node mobility is not specified by the authors. DCA is mostly applicable for a static or a quasi static network. In the
3.5 Works related to MANET Parameters

Distributed and Mobility Adaptive Clustering (DMAC) algorithm [6] proposed by the same authors, the cluster formation process is almost same as that of DCA. However, the non-mobility of nodes during the execution of the algorithm is eliminated here, making it truly mobility adaptive. DMAC claims to be the most suitable algorithm for the cluster formation and maintenance in the presence of node mobility. It starts with the assumption that every node knows its own ID, weight and status in the network as well as the same for its one-hop neighbours. This proves that the cluster head is selected only with the knowledge of its local topology.

Chatterjee et al. [18] included several factors like degree difference, sum of distances, cumulative time of serving as cluster head to find weights of a node. Their proposed weight calculation formula is given in equation 3.6.

\[ W^v = w_1 \Delta^v + w_2 D^v + w_3 M^v + w_4 P^v \] (3.6)

Where \( \Delta^v \) is degree difference and is defined as the difference of degree of a node \( v \) and ideal degree a cluster head can handle. The number of one hop neighbours of a node \( v \) defines its degree. Ideal degree of a cluster head \( CH_i \) is a pre-defined threshold value which defines the maximum number of nodes the cluster head can handle ensuring efficient medium access control (MAC) functioning. \( D^v \) is defined a sum of distances of a node \( v \) from all its neighbours. The next factor \( M^v \) is the running average of the speed of node \( v \). \( P^v \) is cumulative time during which a node \( v \) acts as a cluster head.

3.5 Works related to MANET Parameters

Yu [56] presented a typical ad-hoc network topology and derives the statistical properties of neighbour nodes for that topology. Ngnyen [67] proposed a statistical model to find the neighbour count of a node. Their model is applicable for RWP mobility model only. They do not derive any model for finding neighbours count for other topologies. Our model on the other hand is equally useful for popular mobility models and can be extended for any mobility model.

Link duration is a fundamental parameter when evaluating the mobility in MANET. But
very studies provides a formal description of this variable. Chou and Hayes [24] studied, the mean value of link duration for a single hop case under the constant velocity model. Based on this model, they tried to generalize a model for multi-hop case also but they did not provide any closed form solution for this model. In [33], the authors presented an analysis of link duration for a two hop MANET only. The authors considered that the source and destination nodes are fixed while the intermediate nodes are moving using Random Way Point (RWP) mobility model. But they fail to extend their ideas to a route of several hops. Shridhar and Chan [93] did a detailed study of the link lifetime characteristics associated with three popular mobility models, namely (i) Random Way Point, (ii) Reference Point group mobility and (iii) Manhattan Grid mobility patterns. In their study, they found that link failures of all models are of wear-out type instead of random failures. They suggested that log-normal distribution is a good fit for network having sufficient amount of randomness in their mobility like RWP. They found that Weibull is a good model for mobility models with relative stable links like RPGM with low speed. The gamma distribution is a good fit for networks following Manhattan Grid mobility pattern because movements of mobile nodes are highly dependent on the decisions made at road junctions.

One of the first attempt to model route duration was by Bai [4]. They examined the detailed statistics of route duration considering the different mobility models. Their observation was that under minimum speed and longer routes (more than 4), the time duration of routes can be approximated by exponential duration. They also evaluated the effect of number of hops, transmission range, and the relative speed of the mobility of different models. However, they did not evaluate the fitness of any other distribution. They also did not justify the selection of the exponential distribution. Han [38] used Palm theorem to show that under some circumstances of infinite node density, the route lifetime with large number of hops converge to an exponential distribution. Both these works are valid for routes with large number of hops so these studies cannot be directly applied to many practical MANET applications where the path consists of a few hops only.

Delay prediction plays an important role in the design of the congestion control, routing and flow control, network configuration and distributed communication system. In recent years, queueing theory [26, 51], system identification [69], and time series [68, 100] approaches have been adopted to model the network delay dynamics. Guo at el. [34] presented
3.5 Works related to MANET Parameters

A scheme for predicting mean per-packet one-hop delays using neural network approaches. The predicted one-hop delays are then used by the nodes to participate in the routing information diffusion. They prove the feasibility of predicting mean delays as a time series using either tapped-delay-line Multilayer Perceptron (MLP) network or tapped-delay-line Radial Basis Function network (RBFN) through experiments. They used two types of inputs for prediction: a) the mean delay time series itself only, b) the mean delay time series together with the corresponding traffic loads. Hongyan et. al. [42] used Autoregressive models and neural network to predict Internet time delay. Tabib and Jalali [95] used feed-forward multilayer Perceptron to predict Internet time delays. Guo et. al. [35] proposed a delay prediction mechanism and integrated that prediction mechanism with a proactive ad hoc network routing protocol called Optimal Link State Routing (OLSR). They used queuing delay only, and showed that queuing delay can be modelled as a non-stationary time series. They used Multilayer Perceptron (MLP) and Radial Basis function to predict from the non-stationary time series model of queuing delay in mobile ad hoc network.