ROUTING IN THE PRESENCE OF GROUPS IN MANETS

By

MADHUSOODAN PARTHASARATHY

A thesis submitted in partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

WASHINGTON STATE UNIVERSITY
School of Electrical Engineering and Computer Science

MAY 2009

To the Faculty of Washington State University:

The members of the Committeee appointed to examine the thesis of MADHUSOODAN PARTHASARATHY find it satisfactory and recommend that it be accepted.

____
Chair

____

____

ACKNOWLEDGEMENTS

I thank my adviser Dr. Min Sik Kim for his guidance during the past one year, especially for his reviews of my write-ups and feedbacks on the simulations. I also thank my previous adviser Dr. Murali Medidi for his support during the initial stages of this work, and particularly for helping me choose my thesis topic. I would like to thank Dr. Zhe Dang and Dr. Carl Hauser for serving on the Defense Committee and reviewing my thesis. I also acknowledge my friends at WSU, members of the Network Research Laboratory (NRL), and the staff and faculty of the School of EECS, WSU for the innumerable ways in which they have helped me throughout my studies at WSU.

ROUTING IN THE PRESENCE OF GROUPS IN MANETs

Abstract

by Madhusoodan Parthasarathy, M.S.
Washington State University
May 2009

Chair: Min Sik Kim

A *Mobile Ad hoc NETwork* (MANET) is a collection of co-operative mobile nodes that communicate over a shared wireless medium, in the absence of a supporting infrastructure such as base stations. It is envisaged that MANETs will serve applications in which a group of mobile nodes collaborate to carry out a specific task, for instance, a group of mobile tourists taking a city tour. Mobile nodes participating in a group adhere to a movement pattern that maintains spatial proximity between all group members. The concordant motion of team members collaborating on a specific goal is termed *group mobility*. The coordinated motion of group members and the resulting group-wise clustering of nodes, as well as the atypical traffic requirements of group mobility scenarios present a substantially different networking environment than in conventional MANETs. A routing protocol specifically tailored to exploit group mobility is, therefore, designed in this work. A service discovery-based approach to the creation of groups at the application layer and their configuration at the routing layer is first described. Next, a hierarchical design of the routing protocol that discriminates between intra- and inter-group communications is proposed. A proactive intra-group routing scheme is employed in view of the relatively small sizes of groups, and in anticipation of a healthy percentage of intra-group connections in the overall network traffic. A reactive inter-group routing algorithm is overlaid on the intra-group routing mechanism in such a manner that the propagation of *route request* packets is restricted within certain *directions* in the network. The routing protocol is implemented in *ns-2* and a performance comparison against AODV [1] and DSDV [2] is made under different traffic loads, connection mixes etc.. From a total of 225 experiments, it is observed that the proposed protocol achieves 25% and 30% gains in network throughput over AODV and DSDV respectively.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

A *Mobile Ad hoc NETwork* (MANET) is a collection of cooperative mobile nodes that communicate over a shared wireless medium, in the absence of a supporting infrastructure such as Base Stations (BS) in conventional wireless networks. The nodes in a MANET are self-sufficient in the sense that they do not require the facilities of specialized routers or, generally, of any centralized server for network operations. The network is therefore required to be self-configuring and self-organizing, and concomitantly, rapidly and easily deployable. Freedom in node mobility and the lack of an infrastructure result in impromptu and arbitrary network topologies.

The relatively limited transmission range of mobile devices with respect to the potentially large geographical areas that MANETs might span entails multi-hop communication in the network, wherein each mobile device acts as a packet forwarder. In fact, the deliberate limitation of the transmission ranges of mobile devices is sometimes pursued so as to reduce channel contention in dense ad hoc networks.

The absence of specialized routing devices leaves the onus of data forwarding upon all nodes that reside in the network. In fact, the absence of a supporting infrastructure means that the functionality implied in the various layers of the network stack must be cooperatively managed by all the nodes in the network. Thus, MANETs are characterized by an equitable distribution of power and responsibility amongst the nodes, as against the centralized management of a traditional wireless network by the BS. Such a cooperative engagement requires the selfless endeavor, on the part of any MANET node, to facilitate any network operation irrespective of whether the node directly benefits from it or not.

MANETs are to be readily deployable in unfamiliar environments where no communication infrastructure exists to facilitate wireless networking between mobile devices. MANETs are especially foreseen to play an important role in military communications in battlefield scenarios, and in civilian environments like classrooms and conferences. MANETs were initially expected to serve only battlefield environments. Some reasons for the foray of ad hoc networking into commercial areas are identified in [3] as the proliferated availability of low-cost mobile devices, the emergence of wireless technologies such as IEEE 802.11, W-CHAMB etc., the potential for increased radio coverage of broadband wireless systems with ad hoc networking and the need to extend the multimedia services of the Internet into wireless systems. In the civilian environment, large-scale isolated ad hoc networks are not expected to flourish beyond the measure of academic endeavors (because of problems in imparting security and from limited traffic performance), whilst smaller isolated ad hoc networks are expected to serve domestic and business set-ups [3]. MANETs are also expected to be connected to the mainstream Internet, whenever feasible.

Research work on MANETs principally address 2 issues: the scheduling of access to the shared wireless medium (Medium Access Control or MAC) and the routing of data packets from a source node to the destination through intermediate nodes as relays [3]. Effective scheduling mechanisms are required to control access to the wireless channel by mobile nodes, in a fair manner, so that collisions from simultaneously transmitted data packets may be minimized. Several MAC layer scheduling mechanisms have been proposed (e.g. [4, 5]) as alternative designs to the existing IEEE 802.11 set of standards (and amendments), as well as to adapt MAC scheduling to specific ad hoc environments (e.g. [6, 7]).

The routing layer serves to establish communication paths between source-destination pairs using other nodes lying on the path as forwarders. Node mobility and the erratic nature of the wireless channel result in frequent mutability of connection statuses between mobile devices, leading to unforeseeable topological changes in the MANET. The wireless channel is susceptible to frequent disruptions from channel fading and is an inherently lossy communication medium, thereby compounding upon the difficulties in maintaining a standalone network of mobile nodes. Routing in MANETs is remarkably difficult than wired IP networks because of both, the lack of supporting infrastructure and the resulting hierarchies like subnets, as well as the frequent topological changes that ensue from node mobility and the erratic nature of the wireless channel. Wireless IP makes use of BSs to manage the topological changes in the network; however, a MANET is not equipped with such an organization.

Depending on the nature of the application, MANETs of diverse sizes, from a handful of nodes to tens of thousands of mobile devices, are anticipated. Routing solutions for ad hoc networks must scale well to large network sizes. The increase of network size must not limitlessly add to the size of routing tables at nodes, nor proliferate the amount of routing updates exchanged in the network. High mobility in the network may result in topological disturbances and disruptions in connections, and the network must choose routes that will require minimal reestablishment of connections (for example, based on mobility prediction), or, must be able to quickly reconcile to such changes. Routing protocols must sparingly use the networks resources during the establishment of routes or exchanges of routing tables because the wireless channel is an extremely resource-constrained environment to work on. Further, routing protocols must be designed while bearing in mind that electrical power in wireless devices is a limitedly available resource.

Some reasons why conventional routing protocols (for wired media) are unsuitable to the MANET environment are presented in [2]. They are enumerated here,

1. Conventional protocols cannot provide the kind of dynamic, self-starting behavior needed for ad hoc networks.

2. The highly dynamic network topologies of MANETs are not conducive to conventional routing protocols in light of the slow convergence characteristics of these protocols.

3. Lastly, these protocols place a heavy computational burden on each mobile device (which must operate as a router).

Accordingly, several routing protocols have been designed specifically for the MANET environment, of which a few are discussed next.

Routing solutions for ad hoc networks can be broadly categorized as under reactive or proactive approaches towards route discovery. Proactive routing protocols periodically maintain routes to all or a subset of destinations throughout the lifetime of a source node, irrespective of whether a communication between a source-destination pair is imminent. They are based on either Link State Routing (LSR) or the Distance Vector (DV) scheme, and may include some or all of the nodes in the network depending on the organization of the network structure. Proactive protocols are unsuited to large networks because of network inundation from the periodic broadcast of routing updates of large volumes.

In the reactive approach to route discovery, a route between a source-destination pair is discovered when communication between the 2 nodes is desired, thereby eliminating periodic route maintenance. This approach is used in protocols such as DSR (Dynamic Source Routing) [8], AODV (Ad hoc On-Demand Distance Vector routing) [1], TORA (Temporally Ordered Routing Algorithm) [9] etc. Reactive protocols are more suited to networks where the traffic load is light. It has been shown by simulation that the amount of control overhead generated in a network that uses AODV surpasses the actual data traffic that the network carries, under heavy traffic loads.

The availability of the location information of mobile nodes from GPS technology has been exploited to aid the routing process in schemes like LAR (Location-Aided Routing) [10] and DREAM (Distance Routing Effect Algorithm for Mobility) [11]. In LAR, the positional information of mobile nodes is piggybacked on all communication exchanges. Reactive routing that limits the scope of flooding of *route request* packets to a region where the destination is estimated to be located is performed. Although the protocol's ability to avoid network-wide flooding has been demonstrated, it is also noted that LAR is only applicable to a special environment: where all mobile devices are equipped with a GPS (or some means of localization), and this can not be guaranteed.

MANET protocols that view the network as a hierarchy of mobile nodes, like the subnet organization of IP, have been proposed in [12–15]. Such an organization may also be made to reflect the logical partitions that may exist in the network [12]. The most common technique to set up such a hierarchy is to cluster nodes that are in close proximity and choose a representative node, called the *Cluster Head* (CH), of the cluster. The CHs form a backbone network, and are responsible for all routing updates exchanged in the network. A source that seeks a communication path to a destination uses the routing update sent by the destination's CH to locate the destination for data transfer. The use of hierarchies reduces the routing overhead in the network by requiring only the set of representatives (CHs) to participate in route maintenance. However, frequent node mobility shatters the hierarchy, and the maintenance of the hierarchy itself becomes expensive. Clustering, and other mechanisms to construct hierarchies are suitable for moderately dynamic topologies but may result in a differentiated consumption of network resources by burdening the representatives.

It is envisaged that MANETs will serve applications in which a group of mobile nodes collaborate to carry out a specific task, for instance, a group of fire-fighters involved in a search and rescue operation, a group of aircrafts flying in formation, a group of mobile tourists etc. Members of such teams, in coordinating with one another on a common goal, show similar mobility patterns as one another. Further, depending on the application, such teams may also have a central commanding authority in the form of a *Group Leader* (GL), e.g., a squadron leader of a group of aircrafts. The motion of the members of the group is usually circumscribed to within a certain distance from the GL. The concordant motion of team members that collaborate on a particular goal is referred to as *group mobility*.

While the movement of nodes in MANETs without groups is modeled as independent, MANETs with groups evince a correlation in the mobility patterns of group members. The network in Figure 1.1 contains 3 groups, A, B and C. The GL of each group has been represented by a solid circle in color, while other group member nodes are denoted by the circumferences of circles in color. The bigger circles in black loosely define the physical boundary of the group, which is the polygon obtained from connecting the locations of the outlying nodes of the group. GLs can be located anywhere in the *group area* (the area bounded by the polygon/circle), and need not be centrally positioned in the group. While it is assumed that nodes that collaborate on a functional goal as a group will strive to stick together in a close physical configuration within this circle, it is also noted that at times, certain nodes may drift away from the periphery of the group or the common trajectory of the group, but may later rejoin the setup. The double-edged circles in black depict *individual nodes* which are isolated nodes, unassociated with any group in the network. Individual nodes move independently of any other node in the network (in consistence with node mobility modeling in conventional MANETs). Thus, the network scenario is to include several groups, alongside which individual nodes co-exist.

A group may be regarded as a single, clustered entity of several mobile nodes that move independently of other nodes in the network. The clustered disposition of group members, as well as the presence of GLs consolidates a hierarchy in the network: the higher level being the interaction between groups, and, the lower level of interaction between nodes of a group.

The affiliations of mobile nodes to groups are dynamic: groups may merge and disintegrate; nodes may affiliate and de-affiliate from groups. E.g., a group of search-and-rescue workers may separate to search different sections of the building and then merge again thereafter. Likewise, in-

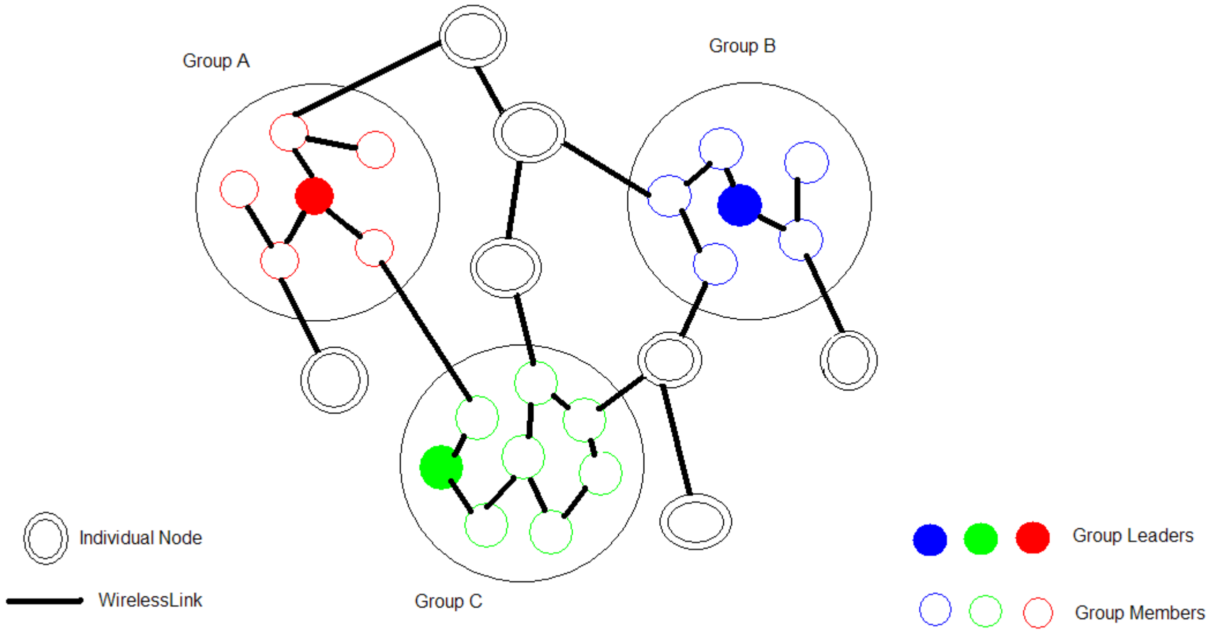


Figure 1.1: Group Mobility.

dividual nodes may join groups, and nodes may de-affiliate from groups and become autonomous. Thus, the number of groups in the network, and, the sizes and constituents of groups change continually.

Communication in a mobile environment with groups can take the form of *intra-group communication*, *inter-group communication* and communication originating and/or terminating from an individual node. Intra-group communication refers to the communication between member nodes of the same group that collaborate on a mission. Inter-group communication refers to the communication between mobile nodes that belong to different groups. The presence of groups is expected to strongly affect the traffic dynamics in a MANET. The functional dependence of mobile nodes in a group to accomplish their mission suggests that a substantial portion of the communications involving group members is concentrated within the group. Inter-group communication to, for example, connect to a server lying outside the group or solicit membership in other groups, will also be desired.

As mentioned before, the motion of group members is circumscribed to within a certain distance from the GL, although the off-and-on drifting of nodes outside the boundary of the group is conceivable. This roughly translates as requiring that all the group members form a connected network amongst themselves. That is, any 2 group members of a group must be reachable through a path which is exclusively composed of nodes that belong to the group. This allows nodes the freedom to move anywhere within the group area (at any particular instant), thereby resulting in moderate topological dynamism within the group. Topological dynamism in groups is also a ramification of the provisioning for group splits and mergers.

The application of other group mobility models like RPGM (Reference Point Group Mobility Model [16]) result in more coordinated group mobility by requiring all nodes to more-or-less mimic the motion parameters of the GL. The velocity vector of each group member is approximated as the modulation of the GL's velocity vector by a random motion vector. The RPGM model reckons a disparity in the relative mobility between nodes of the same group and of those that belong to different groups, pointing to more topological stability in the network formed by a group than in a conventional MANET.

Group mobility of nodes induces a natural hierarchy in the network by clustering nodes belonging to the same group around their GL. The knowledge of a route to a single node of a different group is enough to accomplish route maintenance to any member of that group because of the spatial proximity of the nodes belonging to a group. In that sense, routes to all nodes of a group may be summarized by a route to a representative node of that group [12]. Routing protocols that take a *flat* view of the network topology are not suited to exploit the inherent hierarchy in such MANETs. Routing may therefore be distinguished at 2 levels: intra-group routing that manages the communication requirements between group members, and inter-group routing to serve communication between nodes belonging to different groups. Routing in the presence of group mobility is similar to routing over networks over which artificial clusters have been created.

Coordinated intra-group node mobility, group-wise node clustering and atypical traffic requirements of group mobility scenarios present a substantially different networking environment from conventional MANETs. It has been shown, by simulation, in [17, 18] that conventional MANET routing protocols perform better on group mobility scenarios than when assuming independent movement of nodes in a MANET. In light of the above factors, the development of a routing protocol that facilitates communication in a MANET, in the presence of groups, is the subject of this work. Routing that is specifically tailored to exploit group mobility in MANETs, has already been explored in LANMAR [12] and HSR (Hierarchical State Routing) [19].

Although it was implied that group mobility presents distinct circumstances in which the routing protocol must operate, some qualifications to the assumptions are in order. Firstly, it was pointed out before that a healthy percentage of the network connections would involve members of the same group. This is only suggested as a guideline; the network is expected to function at all load combinations. Next, even though the mobility patterns of group members are predominantly concordant with the GL, nodes may still drift off the trajectory of the GL.

An approach towards the setting up and operation of groups is presented in Section 3. This includes mechanisms to advertise, infer the presence of, and, associate and dissociate from groups. It also covers the supplying of every group with a unique identifier (called Group ID) within the MANET, in order to effectuate a network hierarchy.

The possibility that groups may merge and divide renders a static perception of the group affiliations of nodes in the network inappropriate. A Group ID discovery service is then required to maintain and disseminate information about the current affiliations of nodes, so that the benefits of the hierarchical structure may be fully realized. The Group ID discovery service must be a light-weight mechanism[6] and it may be realized as a part of the routing scheme itself, or may complement the routing scheme.

As noted before, groups in the network may be composed of varying numbers of nodes resulting in small, medium and large groups in the network. This poses a question on whether groups, being specialized sub-networks in a MANET, must be allowed autonomy over the choice of intra-group routing schemes to suit their organizations, sizes and traffic requirements. Even if a group were to be allowed choice over its intra-group routing scheme, it must be cognizant of MANET-level mechanisms to forward inter-group traffic between other groups.

It is assumed that groups will, at most times, form a connected network within the members of the group exclusively. However, the possibility that a node may drift off the groups trajectory allows for physical partitioning of the group (that is the group is no longer connected exclusively through its group members), although, functionally the separated segments may still desire collaboration; whether this has a bearing on the routing protocols operation must be addressed.

The routing solution for MANETs that demonstrate group mobility is designed based on the following criteria,

1. The presence of groups lends a hierarchical organization to the MANET structure; a hierarchically-designed routing protocol will, therefore, be better-suited to such an environment.

2. The choice of the intra-group and inter-group routing schemes employed must be based on both the sizes and topological stability of groups, as well as the anticipated traffic distribution of intra- and inter-group communications.

3. The GL may only be regarded as a facilitator of the group; excessive reliance on the GL by

[6] A network-intensive approach to Group ID discovery is touched upon and ruled out in Section 4.2.

the routing protocol is unwieldy as it creates a performance bottleneck at the GL.

4. The performance of the routing protocol must not degenerate with the presence of individual nodes, that are unaffiliated to any group.

5. As with any routing solution, an optimal balance between *throughput*, *packet delay* and *control overhead* is desired. The routing protocol must scale well to large network sizes: the presence of groups is an incentive to satisfying this requirement.

CHAPTER TWO

RELATED WORK

Since MANETs are cooperative networks that lack any routing infrastructure, routing protocols operate at each mobile device in a manner to facilitate networking between any 2 nodes in the network. Routing protocols vary in their approaches to route discovery and maintenance, their view of the network organization, the network parameters they tend to specifically optimize etc. This section surveys some prominent protocols under the various classifications, and evaluates their applicability to the group-mobility scenario. Also, the end of this section examines the modeling of group mobility, especially from the perspective of computer simulations.

Route discovery and maintenance in MANETs can take *proactive*, *reactive* and *hybrid* approaches. Proactive approaches require that nodes regularly exchange routing information with one another in an endeavor to maintain consistent topology information at all nodes. Routes to any destination node are gathered and maintained at all times, so that a route is always available when a communication need arises. Proactive routing based on the Link State (LS) or the Distance Vector (DV) schemes is popularly employed in the wired Internet with some modifications to prevent routing loops from surfacing in the network. In LS routing, every node periodically floods information about its adjoining links to the rest of the network. A network topology map is constructed at each node, and a shortest path algorithm is applied to determine routes to all nodes in the network. In the DV scheme, nodes periodically broadcast their distance estimates of other nodes in the network, to their neighbors. The neighbor node that is closest to a destination (determined as the bearer of the smallest distance estimate to the destination amongst all neighbors) is chosen as the *next hop* to the destination. The original DV scheme is however plagued by slow convergence and the *count-to-infinity* problem which may result in the formation of routing loops in the network: inter-nodal coordination and synchronization mechanisms are required for their resolution [19].

Destination-Sequenced Distance Vector Routing (DSDV) is proposed in [2] as an adaptation of the DV scheme to the MANET environment. As in the basic DV scheme, all nodes periodically broadcast their distances to the other nodes in the network, to neighboring nodes. DSDV also uses *Sequence Numbers* (SNs), which are basically timestamps associated with every routing update, to eliminate stale information from propagating in the network and in preventing routing loops. SNs denote the freshness of routing information—information with a higher SN overrides that with a lower SN—and are generated as increasing even numbers in successive routing updates made by nodes about themselves. Apart from the periodic broadcasts of routing updates, nodes also send trigerred updates when a significant topological change has occurred in the network to keep pace with the dynamism of the MANET. DSDV also maintains a history of the elapsed time between the arrivals of the first routing update and the update containing the best path received from a destination, as an indicator of the average inter-arrival time. This is used to avoid the generation of spurious routing updates that don't advertise paths that have the best routing metric. In particular, when the routing update containing the best path to a destination is received later than any other routing update from the destination, the maintenance of the average inter-arrival time ensures that repeated advertisements that vacillate between the best and the most current path to the destination are not trigerred.

WRP [20] is a proactive LS-based routing algorithm that broadcasts routing updates only to neighboring nodes. Updates are transmitted whenever nodes detect a change in the status of their adjoining links, or, when an update is received from their neighbors. Sequence numbers are used to resolve freshness between updates. In OLSR (Optimized Link State Routing) [21], every node selects a subset of its neighbors, whose broadcasts can together cover the link states of its 2-hop neighborhood, as Multi-Point Relays (MPRs). Only MPRs are responsible for sending LS updates throughout the network in the form of *topology control* messages, which are sent periodically. 2-hop neighborhood information is derived from an exchange of neighborhood information between neighboring nodes. The amount of information flooded in the network is thus reduced by requiring only MPR nodes to generate and forward LS updates. TBRPF [22] broadcasts LS updates in the reverse direction along the spanning tree formed by the minimum-hop paths from all nodes to the source of the update. Topology control messages, which are LS updates, are sent when a change in link status is detected between a selected parent/child pair in the routing tree of a node. TBRPF uses the topology information received along the broadcast trees to compute the minimum-hop paths that form the trees themselves. Sequence Numbers are used to guarantee the freshness of topology updates.

Proactive protocols are unsuited to large networks because of the enormous control overhead generated from the periodic flooding of global topology information. The volume of routing overhead and the size of routing tables are proportional to the size of the network, making them unscalable to large network sizes. Further, in most proactive protocols, the broadcast of new routing information is not carried out immediately because it can potentially result in a broadcast storm in the network. Instead, routing information is aggregated over time and scheduled routing broadcasts are made by nodes[2]. The lack of synchronization in the timing of routing broadcasts made by nodes delays the propagation of routing information to interested nodes. The staggered propagation of routing information can render the routing information stale when such updates are about nodes that are distantly located.

Reactive routing protocols discover routes between a source and destination, whenever communication is desired, by flooding a *route request* packet to the network. When the destination

[2] Sometimes, routing advertisements may also be trigerred with immediate effect depending on their significance, like in DSDV.

receives the request packet, it sends a *route reply* back to the source on the reverse of the path through which the request reached the destination. Route discoveries are initiated only when nodes require to communicate, thereby precluding a continual and global route maintenance system.

Ad hoc On-Demand Distance Vectoring (AODV) [1] is a reactive descendant of DSDV. Route Request (RREQ) packets are flooded throughout the network by a source requiring to communicate with a destination. When nodes propagate the RREQ packet, they set-up pointers on the reverse path to the source, so that they may forward the Route Reply (RREP) packet back to the source, if need be. When the destination receives the RREQ packet, it unicasts the RREP packet back to the source. The RREP packet sets up the route on the forward path from the source to the destination, with each intermediate forwarder marking down its net hop to the destination in its routing table. Sequence Numbers (SNs) are used to guarantee the freshness of routing information, and to eliminate the possibility of routing loops. In particular, a source that seeks a route to a destination specifies a minimum SN that a route to the destination must meet for it to be acceptable to the source. As a consequence, replies are generated only by the destination or by intermediate nodes that possess a recently learnt route to the destination. It is shown that this strategy prevents the formation of routing loops in the network [1]. When the route between 2 communicating nodes collapses, route re-discovery is initiated by the source upon notification by an intermediate node through a Route Error (RERR) packet. The use of *expanded ring search* during route discovery and the local repair of failed routes at intermediate nodes have been proposed as optimizations to AODV.

Dynamic Source Routing (DSR [8]) has similar route discovery and maintenance phases as AODV but uses *source routing* to forward data packets, unlike AODV which maintains next hop information in the routing tables of intermediate nodes for packet forwarding. The source route is built into the route request packet as the sequence of hops traversed by the packet before reaching the destination. The destination reverses this path and includes it in its reply packet to the source, so that, intermediate nodes forward the reply based on the route contained in it. Data packets are also routed using the source route that is appended to the payload. While the use of source routes in the data packet consumes extra space, it can trivially avoid routing loops and provide caching information to nodes overhearing or forwarding the packet [8].

Reactive routing protocols discover routes only when communications are desired, thereby avoiding the regular flooding of routing updates in the network. However, they incur considerable latency in route acquisition, and do not perform well when the traffic load on the network is heavy. In a network with a large number of connections, the number of route discoveries made by AODV congests the network to a state that more control packets are generated than the actual amount of data transferred. Since proactive schemes continuously maintain routing information at each node, to the rest of the network, they do not incur a route acquisition delay unlike an on-demand scheme; yet, this leads to the wasteful maintenance of routing data between nodes that do not require to communicate with one another.

All the above routing protocols operate upon a *flat* network topology, and are hence, in their native forms, unsuited to exploit the hierarchical structure that group mobility lends to the network organization. Further, route maintenance in the above protocols does not discriminate between the levels of relative movement between nodes belonging to the same group and those of different groups, or between the volumes of intra- and inter-group traffic. However, the 2 broad approaches of proactive route maintenance and on-demand route discovery offer various tradeoffs in the control overhead depending on the mobility and traffic characteristics of the network, and serve as the building blocks of several other routing protocols as detailed below.

Hybrid routing integrates the 2 approaches by performing proactive and reactive routing on different sections of the network. The Zone Routing Protocol (ZRP [23–25]), defines a proactive routing zone of a certain hop radius around each node, where route maintenance is done in a proactive manner. Within this zone, nodes periodically exchange routing updates based on the IARP (Intra-Zone Routing Protocol) protocol, which is a split-horizon version of the DV algorithm. Routes to nodes falling outside the routing zone are discovered reactively, whenever the source node desires a communication. Controlled flooding of route request packets in the Inter-zone Routing Protocol (IERP) is achieved by taking advantage of the pro-active maintenance of routing information within the routing zone. Specifically, when a source node does not know of a route to the destination, it sends the route query packet to all the *peripheral nodes* of its routing zone: peripheral nodes being the nodes which are located at a distance equal to the *zone radius* from the source node. The recipient nodes check for a route to the destination in their routing tables, and in the event of its unavailability forward the query to the peripheral nodes of their routing zone. This process is repeated until a node that possesses a route to the destination receives the query packet, to which, it issues a route reply. The propagation of the route query is thus limited to a subset of the nodes in the network.

However, the overlapping of routing zones of various nodes results in the forwarding of the query packet into routing zones previously covered by other threads of the same route query. Two levels of redundant-query detection are implemented to detect and discard superfluous threads of the same query. At level 1, nodes that have forwarded the query previously to the peripheral nodes of their routing zones detect and discard repeated threads, while at level 2, nodes that have previously overheard a query, owing to the broadcast nature of the wireless medium, terminate newer instances of the same. [25] provides additional query termination procedures called *Early Termination* and *Random Query Processing Delay*.

ZRP is extended in [26] to allow nodes to have independent sizes for their routing zones, adapting them to their traffic characteristics. Known as IZRP (Independent Zone Routing Protocol), the protocol requires that nodes continuously monitor the amount of intra- and inter- zone traffic to derive the optimal zone radius that suits their traffic requirements: higher the intra-zone traffic, bigger the routing zone; higher the inter-zone traffic, smaller the routing zone. IZRP maintains topology information within routing zones of varying sizes by distinguishing 2 kinds of zones at each node: *send* and *receive* zones. The send zone of a node consists of all nodes that require a pro-active update from it, while the receive zone is the normal routing zone that the ZRP uses, from which updates are due. Every node broadcasts a *zone-build* update message to its receive zone, so that, all nodes may construct the member set of their send zones and send their routing updates to those nodes. Although adaptation to traffic characteristics is beneficial, the explicit task of constructing routing zones undoes the simplicity of the ZRP scheme, especially when considering the topological dynamism of MANETs.

TZRP (Two-Zone Routing Protocol) [27], another variant of the ZRP, incorporates nodal mobility characteristics into the sizing of routing zones by defining 2 routing zones for a node. In IZRP, routing zones are sized to reflect the traffic requirements of a node. High mobility may however render routing updates to peripheral nodes inaccurate, thereby affecting the performance of the IERP protocol. TZRP defines a smaller *crisp zone* where proactive route maintenance can be performed accurately to assist the IERP. A *fuzzy zone* that extends beyond the perimeter of the crisp zone is constructed and maintained with a fuzzy-sighted, less accurate protocol like FSLS [28] so as to exploit traffic locality in the network. Reactive routing to nodes beyond the fuzzy zone is performed like in ZRP.

SHARP (Sharp Hybrid Adaptive Routing Protocol) [29] defines *hot destinations* as nodes to which communication needs arise frequently at other nodes in the MANET. Routes to Hot Destinations are proactively maintained at all nodes that lie within a certain hop radius—the value of which depends upon the popularity of the destination—by a protocol that combines the features of DSDV and TORA. A source node will have a route to a destination if it lies within the forementioned zone radius, or else it uses a reactive protocol like AODV to determine a route to the destination. The degree of hybridization, which depends on the zone radius of each node, is made adaptive to traffic characteristics so as to specifically support QOS requirements with respect to delay jitter, packet loss rate and control overhead.

Hybridized routing reconciles to the disparities in mobility and traffic dynamics between different sections of the network by engaging different mixes of proactive and reactive components in the routing scheme. Proactive routing based on periodic route advertisements is more suited to network segments that have low relative mobility and dense traffic requirements. For instance, the routing zone in IZRP is sized to reflect the traffic density in the node's locality: the size of the node's routing zone, in which proactive routing is performed, is increased or decreased depending on the number of nodes in its vicinity that have communicational requirements with it. In hybrid routing, the fragmented application of proactive routing to different network segments results in a reduction in the size of routing updates. For instance, the size of routing updates within the routing zone in ZRP is proportional to the zone radius (which is much smaller than the network radius), thereby making proactive routing more sustainable within the routing zone. Reactive routing is preferred when connection requirements are relatively fewer, and inter-nodal movement is too frequent to be adequately secured by a periodic broadcast-based proactive protocol. Additionally, in hybrid routing, the construction of routing zones induces a routing structure onto the network whereby the propagation of route request queries in the IERP is deliberately confined to certain *directions* in the network, whilst ensuring network-wide coverage. This reduces the control overhead associated with the route discovery phase of the reactive IERP.

The availability of positional information of mobile nodes from systems like the GPS (Global Positioning System) is exploited by routing protocols like Location-Aided Routing (LAR [10]) to limit the amount of control messages required to track mobile devices in a MANET. LAR uses a reactive approach to route determination, in which, the flooding of route request messages is contained to a *request zone* where the destination node is expected to be placed. The request zone is approximated based on the previous location and mobility values (such as velocity) of the destination, which the source has learned from the information piggybacked on all data packets exchanged between the 2 nodes. In DREAM [11] nodes periodically exchange control packets containing their location information. The rate of exchange of location information is higher to closer nodes than nodes located farther away. Using the location information of the destination, the source sends the data packet in the direction of the destination, such that it is forwarded to the destination with increasing route accuracy by subsequent intermediate nodes. The availability of locational information of mobile devices can not be guaranteed under all circumstances, and is not assumed in the design of the routing protocol in this work.

Global State Routing (GSR [30]) is an adaptation of the LS scheme to the MANET environment. In GSR, all LS updates are sent on a periodic basis so that all changes in link statuses that happen prior to the update timepoint are aggregated and broadcast in a single message. Nodes use sequence numbers to purge stale information from their routing tables. A multi-scope routing scheme called FSR (Fisheye State Routing) is proposed in [13], wherein, nodes send LS updates about nodes at different distances from them, at different frequencies to their neighbors. The use of global-topology updates in GSR results in large messages. As a remedy, FSR sends updates about farther-lying nodes at a lower frequency to a node's neighbors. Therefore, routes to distant nodes are not accurate to start with, but as a data packet makes its way into the network, it is progressively redirected by intermediate nodes in the direction of the destination. Thus, reduced routing overhead from differential updating of link statuses is achieved at the cost of routing along longer paths to far-lying nodes in the network.

Clustering is a popular technique to bundle similar nodes together, and create an artificial hierarchy in the MANET by choosing a representative node called a *Cluster Head* (CH) for each group thereby formed [31,32]. The process may be recursively applied to nodes at the highest level in the hierarchy, to create multiple levels of clustered nodes in the network. When applied to facilitate the routing process, clustering generally targets forming groups from nodes that are in close proximity to one another, whereby it constructs an artificial network structure with the objective of reducing the control overhead required for routing. Some routing schemes proposed for clustered networks are discussed next, followed by the advantages and disadvantages of clustering.

[33] aims at reducing the number of forwarding nodes that might have to participate in a broadcast scheme for a MANET, by partitioning the network into 1-hop clusters. Each cluster has an elected CH which chooses a set of nodes from its cluster that bear a direct link to neighboring clusters, as *gateways*. A node that shares a link with a node belonging to another cluster is termed a *border node*. The CH chooses only one border node to act as a gateway to a particular neighboring cluster. Thus, all the other border nodes that lead to that cluster are not gateways. In order to guarantee network-wide broadcast coverage, a broadcast packet is required to be re-broadcast by all CHs (so that the packet reaches all nodes within the cluster), all gateway nodes (for the packet to reach all neighboring clusters), and by all border nodes that receive traffic from a neighboring cluster (since it is not necessary that an elected gateway node to a cluster will receive a broadcast packet from that cluster). Further, each border node that receives a broadcast packet from a neighbroing cluster waits for a random time period (called *random assessment delay*) before it attempts re-broadcast the packet. If it hears of the same broadcast from another node during this time, it refrains from its own broadcast; otherwise it performs a re-broadcast of the packet.

In [34], clustering schemes to create and maintain *cliques* (clusters in which all member nodes are neighbors) in the network are proposed along with a routing protocol that operates on the clustered architecture. Nodes that share a physical link with a node belonging to a different cluster are defined as *boundary nodes*. From the construction of cliques in the network, the boundary nodes form a connected network within themselves. Therefore, in order for a routing update to be broadcast throughout the network, it is only required that all boundary nodes re-broadcast routing packets; all the other recipients of routing updates simply use the information to fill in their routing tables. Accordingly, boundary nodes send periodic routing updates consisting of the names of nodes belonging to their clique, and the boundary nodes leading to other cliques in the network. With the propagation of this information by other boundary nodes, each node in the network knows the clique membership of a destination and the boundary node that leads to that clique on the shortest path. Thus, the construction of cliques limits the participants in the broadcast of routing updates to boundary nodes.

The Distributed Dynamic Clustering Algorithm (DDCA) for the (*Alpha,t*) cluster framework is proposed in [15]. Additionally, a dual-hybrid routing strategy based on the clustered organization is described. Route maintenance is performed at 2 levels in the network. At the lower level, all nodes pro-actively maintain routes to every other node within their clusters, and to the nearest node of every neighboring cluster. Route maintenance at the higher level involves periodic routing updates sent to all nodes in neighboring clusters by any one node of every cluster. This routing update carries routing information to all clusters in the network. From this exchange, all nodes know of routes to nodes within their clusters and to all clusters in the network. When the destination node belongs to the same cluster as the source, the source uses the route it has learnt from the lower-level routing updates. When the destination belongs to another cluster, the source node queries all clusters in the network to determine the cluster to which the destination belongs. The reception of the query at the cluster in which the destination belongs triggers an affirmative response about the destination's membership in the cluster. The source subsequently routes data packets to that cluster.

Hierarchical OLSR (H-OLSR) [35] extends the OLSR protocol to operate in a heterogeneous network in which some nodes are connected through a separate network of a higher capacity. Each node in the network may be equipped with one or more interfaces resulting in the formation of multiple networks of different capacities among the collection of nodes. A nodal hierarchy is assumed wherein nodes at the lowest level have only a single interface that leads to a low-performance network. At any higher level, each node has multiple interfaces which include interfaces to all networks formed between nodes at lower levels, and to a high-capacity network to which they and nodes situated higher in the hierarchy are privy to. Thus, each network includes nodes situated at or above that hierarchical level. H-OLSR forms clusters in each of the formed networks, the CH of which is any node that also participates in a network at a higher hierarchical level. Nodes maintain topological information about members within the same cluster using the proactive protocol OLSR. Furthermore, at higher levels, each node maintains the membership information of the lower-level clusters represented by the nodes in its own cluster. When a source node needs a route to a destination, it checks if the destination belongs to any of the clusters it participates in, or if the destination belongs to one of the lower-level clusters represented by a node in its own cluster. If so, it routes the data packet to it; otherwise, it forwards the packet to a higher capacity node, which will determine a route to the destination based on the same scheme. Thus, data packets traverse up-and-down the hierarchy depending upon the cluster memberships of the source and the destination. The amount of routing information maintained at a node increases with its position in the hierarchy: the nodes at the highest level maintain routes to all nodes in the collection. The efficacy of such a collection of heterogeneous nodes will depend on the physical distribution of high-capacity nodes in the network: a uniform placement is required to cover the communications originating at all sectors of network .

The partitioning of the MANET into clusters is akin to the concept of subnets in the Internet. While *hierarchical routing* has been successfully applied to the wired Internet, the rapid mutability of the MANET topology may necessitate frequent engagement of re-clustering mechanisms to accurately maintain cluster memberships in the network. Therefore, when relative nodal mobility is high, the overhead incurred by the clustering protocol to reorganize the network may offset the savings from hierarchical routing. In more stable MANETs, hierarchical routing on the clustered network is less expensive since it fragments the amount of routing knowledge that is required to be maintained at each node. Typically, each node maintains routes to members of its own cluster, and depending on the actual specifications of the protocol keeps track of the CHs of the other clusters.

Group mobility ingenerates a hierarchical clustered structure within the MANET: the groups of mobile nodes that move together form well-defined clusters, which are less susceptible to fracture from node mobility than clusters deliberately created from independently-moving mobile devices. The presence of well-defined and stable clusters, as well as GLs to coordinate the application-level tasks of the group make a strong case for hierarchical routing in MANETs that contain groups. However, as noted before, the provision for groups to split and merge, as mandated at the application-level, leads to occasional re-partitioning in the network that the routing mechanism must be wary of.

Landmark Ad Hoc Routing (LANMAR [12]) is specifically designed for large scale ad hoc networks in which subsets of nodes exhibit group-oriented behavior. [36] provides LANMAR with a distributed procedure to elect a group leader (also called *landmark*) for the group.

LANMAR is composed of 2 levels of routing: intra-group routing between members of the same group, and inter-group routing to nodes that fall in other groups. In the basic LANMAR scheme, nodes within a group maintain routes to one another using a modified version of FSR. A *fisheye scope* is defined around each node as containing all group members whose topology information will be exchanged by the node in its routing update. Each node sends a periodic update of the topology within its fisheye scope to its immediate neighbors. This accomplishes proactive route maintenance between any 2 members within a group. Inter-group route maintenance is accomplished by requiring all landmark nodes to flood network-wide distance vector updates periodically. These distance vector entries are also included in the periodic updates made by nodes, and from their network-wide propagation, each node knows of a route to every group in the network. In order to communicate with a destination that belongs to a different group, a source must first determine the group to which the destination is affiliated with. In [12], it is assumed that the affiliation of a node with a group remains static throughout its lifetime, and that the address of a node contains the identifier of the group it is affiliated to. Accordingly, when the destination does not belong to the same group as the source, the source simply parses the identifier of the group in which the destination node belongs and sends the data packet towards the landmark of the target group. As the data packet reaches the periphery of the target group, it is re-routed to the destination by intermediate nodes, thereby circumventing a transit through the landmark node.

[37] extends LANMAR to support dynamism in the group memberships of network nodes: groups may split and merge depending on application-level requirements. Additionally, it is assumed that the addressing mechanism will not reflect the group affiliation of a node in its MANET-level address. When the destination belongs to the same group as the source, the intra-group routing scheme provides a route for the communicational exchange. Otherwise, the source needs to determine the identifier of the group that contains the destination. To accomplish this, the source queries all landmark nodes in the MANET. The group that contains the destination replies to the query, upon which a communication path is established. Thus, even though inter-group route maintenance to landmark nodes is performed proactively in LANMAR, the dynamism in the

group affiliations of member nodes makes a follow-up reactive procedure to ascertain the group membership of the destination necessary. Inter-group routing in LANMAR thus suffers from similar route acquisition latency as a reactive routing protocol, although the extent of propagation of the query packets is contained by the pre-acquired knowledge of network topology. The control overhead in proactive route maintenance in LANMAR is reduced tremendously by requiring only updates from landmark nodes to be network-wide. Propagation of routing advertisements about other group members is restricted to within the group. Yet, as mentioned before, proactive routing based on periodic updates is unsuited to situations where the communicating pair of nodes are at a considerable distance from one another (because of the non-immediacy of information propagation). Therefore, advertisements about landmark nodes may become obsolete when they reach distant nodes in the network.

To accommodate large groups in the MANET, [38] extends LANMAR by partitioning a group into several sub-groups. A landmark representative is defined for each sub-group to act as the leader of the sub-group. FSR is used for proactive route maintenance within the sub-group. All landmark nodes (including the landmarks of sub-groups) broadcast a network-wide distance vector update as part of inter-group route maintenance. From these updates, a node learns and maintains routes to all landmarks of sub-groups belonging to its own group, and to the nearest landmark of all other groups. When the destination lies within the same sub-group as the source, a route is available in the source's routing table. When the source desires to communicate with a destination that lies outside its sub-group, the source parses the *group ID* of the destination from its address and determines the route to the nearest landmark of the destination's group from its routing table. The data packet is then forwarded along this route to the landmark. If the destination lies within the sub-group represented by the receiving landmark, it is forwarded to the destination by either the landmark or any other node in its sub-group. Otherwise, the landmark multicasts the data packet to other landmarks in its group so that the data packet reaches all sub-groups in the destination's group, and eventually, the destination. This is followed by a path-setup procedure that caches the route to the destination at the landmark node that initially received the data packet to obviate the multicasting of subsequent transmissions. When the MANET contains large-sized groups, the partitioning of the group into sub-groups reduces the amount of routing information to be maintained at each node to only the members of its sub-group. The merit of the application of partitioning will however depend on the mobility pattern of the nodes in the group: when intra-group topological stability is low, re-partitioning may have to be invoked frequently.

Another routing solution that accounts for group mobility in a MANET is proposed in [14]. It views the network along 2 dimensions: firstly, as a collection of nodes, of which some functional groups exist; and, as a collection of nodes which may be clustered at different levels based on geographical proximity. Clustering sets up several levels of hierarchy in the network, wherein each node is required to maintain a route to all nodes that belong to the same cluster, at any level in the hierarchy. Groups are thought of as spanning one or more clusters and have unique identifiers throughout the MANET. Every node in the network may be specified by a combination of its group ID and an address that is unique within its group. Each group also has a *home agent* which registers itself with nodes at the upper levels of the clustered hierarchy. Communication routes to nodes within the same cluster are deduced from the topological updates sent within the cluster. When a source node can't determine a route to the destination from its own routing table, it queries its CH for a route to destination's group. The CH looks for a route to a home agent that has the same value of the group ID as the destination node, and routes the request packet to that node. The packet is then forwarded to the destination by the home agent of its group, and a communication path is set up between the source and the destination. The network-wide clustering scheme is, however, oblivious to the natural clustering of mobile nodes that form groups; it may thus cluster together nodes that belong to different groups which creates unstable clusters in the network. Further, the group ID of a node is assumed to be unchanging throughout its lifetime which makes the scheme unsuitable for applications where group memberships are dynamic.

The performance of ad hoc protocols is evaluated by simulations of the protocol's operation on software-generated mobility and traffic patterns. Mobility models used in simulation must mimic motion patterns of mobile devices in real-world scenarios. Traditionally, mobility models used in simulations simply model nodes as unrelated entities moving randomly and/or independently (e.g., Random WayPoint (RWP) and Random Walk Mobility (RWM) models). The prospect of ad hoc environments containing mobile groups has led to the development of several group mobility models, of which a few are discussed next.

The Reference Point Group Mobility model (RPGM [16]) is an extension to the RWP model to capture group-oriented mobility of nodes in a MANET. Each group has a *logical center*, and a geographical radius within which group members are uniformly distributed. The movement pattern of the logical center is specified by a sequence of *checkpoints* that the logical center must pass through at specific time instants. Checkpoints are so designated that group velocity is reflective of the application scenario. Each group member is assigned a *reference point* that moves in tandem with the logical center. A group member's motion pattern is specified as the velocity vector of the reference point modulated by a random velocity vector. Since the motion vector of each group member is closely tied with that of the logical center, the RPGM mobility model leads to a highly correlated intra-group node mobility pattern.

The Structured Group Mobility Model (SGMM [39]) allows flexibility in capturing group dynamics such as group mergers and divisions. Each node or sub-group's position within the group is specified by its distance and angular orientation with respect to the group's leader. Appropriate assignment of these parameters is used to model group mobility and dynamism in group memberships. Another mobility model, based on social network theory, is proposed in [40] in which the mobility patterns of mobile nodes reflect the socialistic tendencies of the users. Other mobility models like [41] include parameters like *nodal acceleration* and *correlation indices* to gauge the affinity between nodes in behaving as a group.

Mobility metrics are quantitative and qualitative measures of the absolute and relative mobility between nodes, and of their impact on the connectivity graph of the network. They serve as a distinguishing yardstick between different mobility models by translating physical mobility parameters of nodes into connectivity measures of the network. They may be used in conjunction with group mobility models to represent the disparateness in the relative mobility between nodes belonging to the same group from those of different groups.

The *degree of spatial dependence* is used in [17] to measure the similarity in velocities between proximate nodes in a MANET. Such a metric that directly measures the mobility characteristics of nodes is termed a *direct mobility metric*. Other direct metrics proposed in the same paper are the *degree of temporal dependence* (the extent of similarity of a node's velocities at two immediate time-points) and the *average relative speed* between nodes. *Indirect mobility metrics* such as *average link change rate*, *average link duration* and *path duration* describe the impact of relative mobility between nodes on the network's connectivity graph. The differentiated performance of a protocol when simulated in different mobility scenarios is explained in terms of the impact of indirect metrics on the building blocks of the routing protocol. [18] compares some connectivity graph metrics of the RWP, RPGM and Freeway mobility models, and reiterates the common observation that group mobility models show greater link sustenance between group members leading to the better protocol performance measurements on group mobility models in [42] and [43].

CHAPTER THREE
GROUP CONFIGURATION

The formation of groups is the result of commonality in the functional interests amongst a set of nodes, which is dictated by the nature of the application. Those application-level groups that result in spatially proximate group members are of particular interest in this work. The routing protocol requires an agency to infer the collectiveness of the nodes belonging to a group, for, the network layer has no bearing on the formation and constituency of groups. To this end, each group in the network is specified by a unique identifier across all groups in the MANET called the *Group ID* (GID). The GID of a group is shared by all its members; when 2 nodes have different GIDs, they reside in different groups and hence do not exhibit any functional dependencies. The assignment of GIDs thus provides for a network layer addressing scheme to distinguish between the groups of a MANET, thereby allowing the routing protocol to infer whether it must operate at an intra-group or an inter-group level when it routes messages between a pair of communicating nodes.

In this section, the procurement of GIDs, including the mapping of the functional assemblage of the group onto a shared GID in the routing layer, the splitting and merging of groups, and the detection and remedying of GID duplication in the MANET from network partitioning and healing, are explored.

The existence and operation of groups in the network may be thought of as a service that is utilized by more nodes depending on their application-level interests. The existence of a particular service in the form of a group must be advertised to the rest of the MANET so that other nodes that foresee a benefit from collaboration with the group may affiliate with it. Then, every single node in the MANET may be regarded as a group in its nascence, to which other nodes may potentially affiliate with.

In order to participate in the routing procedures of the network, each node must possess a GID. This is required of a node even before it actually joins a particular group and takes up the group's GID. The address space of the GID may hence be as large as the MANET address space (when no nodes in the network have formed a group yet). Therefore, before any particular group affiliation is sought by a node, its GID is assigned as the MANET address it has procured from the network. MANET address assignment is expected to be a distributed, unobtrusive and light-weight mechanism to assign unique addresses to mobile nodes in the face of erratic wireless channels and frequent node mobility. Several MANET address assignment schemes have been proposed, e.g., [44–46].

Upon the procurement of a GID (and a MANET address), a node may advertise its position as a potential collaborator to the rest of the network. The purpose of these advertisements is twofold. It serves as a *service discovery* mechanism thereby allowing groups to be formed in an ad hoc manner. For instance, a mobile user may listen to an advertisement of a *city tour* and decide to join the group that is already on it. The other purpose of a service advertisement is to provide a communication path back to the advertiser from the recipient of the advertisement. If the recipient of the advertisement chooses to avail the service, it communicates with the advertiser and moves physically close to the location of the group.

The contents of the service advertisement must in some manner describe the nature of the application that the advertiser wishes to collaborate on, and requires application-level intervention in both advertising and in the interpretation of advertisements. The need for a collaboration can only be specified by the application layer; if a node does not see any point in forming a group it will not advertise itself or respond to advertisements. From service advertisements, nodes also learn of the address of the GL (which is the same as the advertiser's MANET address).

Service advertisements may be piggybacked on routing advertisements made in the network. Hence, the service advertising procedure for groups is closely tied to the routing mechanism of the MANET. Service advertisements may be done proactively by all groups, or, a node that wishes to learn of all services available in the network may solicit service advertisements from the rest of the network. Proactive inter-group protocols like LANMAR [12] may send group advertisements as part of the periodic routing update sent by GLs to the rest of the network. As explained later, the inter-group routing scheme in this work is reactive. Therefore, whenever a node wishes to learn of other groups or potential collaborators in the network it solicits advertisements by sending a request to all groups in the network. In response, all groups and individual nodes that deem a collaboration prudent send back service advertisements.

The formation of groups reflects some need for collaboration amongst its members. Only nodes that see a potential collaboration with other nodes in the network make service advertisements (or solicit advertisements): this is determined at the application layer. Sometimes, knowledge of imminent group formations in the network may be available in advance at some nodes. Consider, for an example, a set of aircrafts flying in formation. They will usually have a *squadron leader*. In such a situation, it is known in advance that the aircrafts will fly together as a formation under the command of the squadron leader. Then, only the squadron leader is required to make a service advertisement; the other aircrafts refrain from such advertisements because, clearly, they are looking to join a group of which the squadron leader is the GL. Also, when concrete knowledge of group formations is available beforehand, it is less expensive to have the GL send out a service advertisement than to have other group members send out solicitations for service advertisements, even if the underlying routing protocol is reactive in nature. In summary, a node that finds it prudent to form groups will procure a GID and advertise its position as a potential collaborator in a group, of which it is a GL.

From listening to the advertisements that circulate in the network (or from solicited advertisements), nodes learn of all groups present in the network, and may choose to join the one that the application layer dictates. The application layer at a node may decide to join one or more groups in the network; but, so far as the routing layer is concerned, such a group join is effected only when the node is physically connected by a wireless link to the said group(s). Accordingly, the application layer first intimates the routing layer of an intent to join a named group in the network. Then, the routing layer inspects the availability of a direct wireless link to the group, whereupon the node may use the group's GID as its own GID and is formally affiliated with the group at the routing layer. The availability of a wireless connection to a group can be detected from the HELLO broadcasts made by its group members as part of local routing procedures (as will be evident from the description of the routing scheme). Should such a wireless link to the said group be unavailable, the routing layer may provide feedback to the application layer of the situation so that the node's mobility trajectory may be altered towards the general vicinity of the host group. Under such a circumstance, the node is not a part of the group at the routing layer: it continues to use its own GID. If a node belongs to multiple groups at the application layer, it may technically belong to many groups at the routing layer too, if it has direct wireless connectivity with all groups. The provisioning for either one or multiple GIDs at the routing layer is upto the routing protocol in question. The use of multiple GIDs in routing messages increases the routing overhead, and is not significantly beneficial than a single GID when the number of instances at which nodes are expected to simultaneously affiliate with multiple groups at the routing layer is small. Therefore, in this work, each node is allocated only one GID, that is, the node uses the GID of one of the groups and participates in its intra-group routing process; at all other groups of which it is a member at the application layer, the node is regarded as a non-member at the routing layer.

When a node wishes to leave a group, the application layer apprises the routing layer of such an intent. Then, the routing layer must change its GID to its own MANET address (or maybe the GID of a different group if the application joins a new group) and detaches itself from the routing mechanism of its former group.

Once groups are formed, the exchange of mobility information, trajectories etc. between the GL and group members may be required to sustain the physical proximity of group members. Such details are beyond the scope of this work. If a node does veer away from the group's trajectory thereby disconnecting itself from the group, then, the continued usage of the group's GID at the node will obfuscate the routing procedures. The routing layer at the node must then relinquish its GID and use its own MANET address as a GID to function as an individual node in the MANET. The disconnection from the group may be perceived from the loss of a physical path from the node to the GL of the group. As described later, the intra-group routing scheme in this work maintains routes between all group members in a proactive manner. When such a route to the GL becomes unavailable for a certain time period, the node may declare itself as physically dissociated from the group. Consequently, the node uses its own MANET address as a GID for routing procedures and may provide feedback to the application layer about the deviation in its trajectory.

The independent mobility of nodes and groups can result in the partitioning and merging of MANETs. The merging of MANETs may result in the presence of duplicated MANET addresses in the network (and hence duplicate GIDs). [44] provides a scheme to detect and remedy address duplication in a MANET that may result from network partitioning and merging (termed *healing* of partitions).

The bottom-line is that some application-level direction is needed for the inference of groups at the routing layer, for, the physical existence of a group is after all a ramification of application-level collaboration. It is noted that the routing protocol does not mandate the physical positioning of mobile devices that form a group; rather, the routing layer discerns the assemblage of a set of nodes through cross-layered exchanges with the application layer.

CHAPTER FOUR
ROUTING SOLUTION

Communication in MANETs that consist of groups is composed of intra- and inter-group communication. Intra-group communication refers to communication between 2 nodes belonging to the same group, whereas the communication between nodes belonging to different groups is termed inter-group communication. Routing mechanisms to facilitate the two forms of communication are different owing to the incommensurate levels of topological changes they must handle and the traffic loads they must support. The intra-group routing mechanism is described next.

4.1 Intra-group Routing

The movement of nodes as a physical group results in a network of nodes in which each member is connected to every other group member without the mediation of a node belonging to another group. Occasionally, however, some nodes may drift outside of the circumference of the group, or lose wireless connectivity (due to the lossy nature of the wireless medium), thereby becoming temporarily disconnected from the rest of the group. Nodal movement, which may be restricted to within a certain distance from the GL, can result in the disruption and in the formation of wireless links between different pairs of group members. Thus, although a stable connected network within the group is anticipated, it is also noted that the formed network will have a dynamic topology from nodal movement. Furthermore, the provisioning for group splits and mergers places an additional requirement on the routing protocol to continually update its knowledge of other members of its group. The intra-group routing mechanism must thus be able to identify and locate other group members at any node in a timely manner.

Ideally, a group must be allowed autonomy over its choice of a routing mechanism to support intra-group communications depending on its size and traffic requirements. However, if the design of the inter-group routing mechanism is closely tied to that of the intra-group routing mechanism (which is reasonable since every inter-group communication path is a concatenation of intra-group segments), then the intra-group routing scheme must not impede the performance of the inter-group routing protocol.

The use of a proactive routing scheme for intra-group routing is proposed here in view of the size, mobility pattern and traffic requirements of the members of a group. Proactive routing protocols require the periodic (in some schemes, event-driven also) exchange of routing information between nodes so that each node has a timely map of the topology of other nodes in the network, irrespective of its communicational requirements. Proactive periodic flooding-based protocols are regarded as unwieldy when it comes to large network sizes because of the large overhead in exchanging routing tables. The relatively small size of a group when compared with the network size translates into smaller routing tables, and hence smaller routing updates, thereby making a proactive routing scheme more tenable for groups.

The formation of specialized groups in the network to collaborate on a specific task suggests that a considerable portion of the traffic load will be intra-group based. Proactive route maintenance is more suited to network segments (such as a group) within which frequent communication requirements arise [26, 27].

The mobility of nodes in a group is circumscribed to within a certain geographical distance from the GL. The mobility patterns of the various group members is however assumed to be independent: they may move to anywhere within the group area (the area swept by the geographic radius of the group about the GL). The unrelated mobility patterns of different group members leads to a RWP-type (Random Waypoint Mobility) mobility modeling of group members within the group area, and is employed in the simulation of group mobility in this work. Therefore, the topological dynamism that a group may be subject to is similar to that of a conventional MANET that does not contain groups.

Other types of mobility modeling for group members are also conceivable. For instance, all group members may strive to mimic the movement parameters of the GL except when radical changes in the physical positioning are required; this may lead to mildly/moderately dynamic topologies which require less frequent routing exchanges between group members. Such a model is not considered here.

Several proactive routing schemes have been proposed: DSDV, WRP, TBRPF, FSR, GSR etc. DSDV has been chosen as the proactive intra-group routing scheme because of its simplicity, and its suitability to the small-sized networks that groups represent. The implementation of DSDV realized in this work is described next.

Every node periodically broadcasts its routing table to the nodes in its neighborhood. This broadcast serves as a *HELLO* message to enable detection of new neighbors and in the exchange of routing information between neighbors. The broadcast message carries the address and GID of the sender, as well as its routing table (or a portion of it). Intra-group routing updates are only exchanged between nodes that belong to the same group. When a node receives a broadcast packet from a node belonging to a different group, it has no use for the routing information contained in the packet. When the broadcast is from a node belonging to the same group, the receiver uses the routing information to populate its routing table based on the routing algorithm. The broadcast also serves as a means to enable the detection of new neighbors, and in the assertion of the continued presence of a node in the neighborhood. Conversely, when a node has not heard a routing broadcast from a known neighbor for a certain period of time, it infers the severance of the link to the former neighbor.

In DSDV, nodes periodically broadcast their entire routing tables to their neighborhood.

Termed a *full dump*, this procedure is repeated only at a nominal rate since it carries a huge amount of information. At a more frequent rate, a portion of the routing table carrying the most important routing information (as deemed significant by the routing algorithm) is broadcast by each node in the group; this is called an *incremental update*. Full dumps are sent very infrequently by DSDV because the large size of the routing tables overwhelm the network capacity, and a sizable portion of the routing tables contain insignificant changes in information since the previous broadcast. The routing table at a node contains an entry for every other node in the group that the node has heard an update about. The most important fields in the routing table are: *Destination*, *Next_Hop*, *Metric* and *Sequence_Number*. The Next_Hop field refers to the address of the neighbor that has the best path to the destination, known from the exchange of routing information. The Metric field denotes the cost of the path between the source and the destination through the Next_Hop node. It is usually a measure of the number of hops between the source and the destination on the route. Out of all the paths available between a source and a destination, the source strives to maintain the route with the best metric (least value when the metric is *number of hops*). Other metrics such as bandwidth, delay etc. may also be used.

When a node makes a routing broadcast, it attaches a value called *Sequence Number* (SN) to the routing table entry corresponding to itself. The SN is incremented by 2 every time a node makes a routing advertisement, and is used to denote the freshness of the update; a routing update with a higher SN is fresher, and hence more credible than an update with a lower SN. Each node in the group maintains a SN which it issues to the routing update entry about itself. Every routing table entry has a SN (filled into field Sequence_Number) which is the value issued by the node in the Destination field in the entry at the time the broadcast was made by that node. The routing updates broadcast by nodes contain the entire routing table or a portion thereof. Each entry in such an advertisement carries the four fields mention above.

When a node detects a link failure to one of its erstwhile neighbors (from the non-arrival of a broadcast from the node for a prolonged time period), the route to the neighbor and to all nodes to which the neighbor served as the next hop are invalidated in its routing table. Invalidation of a route is denoted by incrementing the SN associated with the table entry by 1. Therefore, odd SNs denote an invalidated route, whereas even SNs are associated with valid routes.

When a node hears a routing broadcast from its neighbor, the entries in the advertisement are juxtaposed against their corresponding entries in the node's intra-routing table (by the label of the destination). An incoming routing update overrides its corresponding routing table entry if it has a higher SN. This is in keeping with the semantics of SNs: higher sequence number denote fresher information. If the advertisement and the routing table entry have the same SN, the intra-routing table will carry the entry that has the lower metric of the two. Further, if a route invalidation to a destination is advertised in an incoming update and the receiver has a valid route of a higher SN, then the receiver broadcasts its corresponding routing table entry immediately. This is done to assist neighboring nodes with acquiring a valid route to the destination.

The invalidation of a route results in an immediate broadcast of the corresponding routing table entry. Event-driven broadcasts are also made when a node receives an advertisement that changes the route in its intra-routing table to a destination(s). A change in the metric of a route, a change in the route to a destination and the availability of a route to a destination which was previously invalidated constitute significant changes that warrant an immediate broadcast by a node. It is noted that event-driven broadcasts are costly when they are propagated repeatedly because they can create a storm of broadcasts within the group. Other measures must be taken to limit the propagation of event-driven broadcasts as explained later in conjunction with Figure 4.1.

Based on the changes incorporated to the table, from the processing of a routing advertisement, the node schedules its subsequent routing broadcast; that is, it may be triggered with immediate effect, or, if the changes are not signifiacnt enough they are deferred until the next scheduled broadcast.

The intra-group routing mechanism also schedules full dumps to be broadcast when a certain number of new neighbors have been detected at a node since the last broadcast of a full dump. When a new link to a node is detected, this usually implies a nodal movement that has brought one/both of the nodes to a new sector of the network (group) area. In that event, considerable changes in the routing table of one/both of the nodes are expected. In the absence of full dumps, it will take the node a long time to assimilate routing information pertaining to the new environment from the incremental updates broadcast in the surrounding region. The scheduling of full dumps at such a time allows the node to become acquainted with the routes in the new surroundings in a quicker manner. Full dumps are broadcast out of schedule only when a certain number of new neighbors are detected since the previous full dump (this number is set experimentally).

The lack of synchronization in the time points at which nodes publish their routing tables allows a routing update from a destination to arrive earlier along a longer path than a shorter path.

In Figure 4.1, node *k* has 2 paths to destination *j*: one of length 3 through *d* and another of length 4 through *e*. Let's assume that after the routing broadcast made by *j* the relative time order of broadcasts made by the other nodes are $b < c < a < e < d$; *b* sends its broadcast first, and *d* the last. Then, node *k* will receive a routing update along the shorter path $j \to c \to d \to k$ later than along the longer path $j \to b \to a \to e \to k$.

In these circumstances, the fact that node *j* accepts an update with a higher SN irrespective of the associated metric means that it will send a triggered update when it receives a route through *e* of length 4 that has a higher SN than in its table. Then when *j* learns of a shorter route through *d* of length 3 which has the same SN as the entry in its own table, it sends another triggered broadcast to cope with a route change. If this pattern of broadcast timings within *a, b, c, d* and *e* persists, then

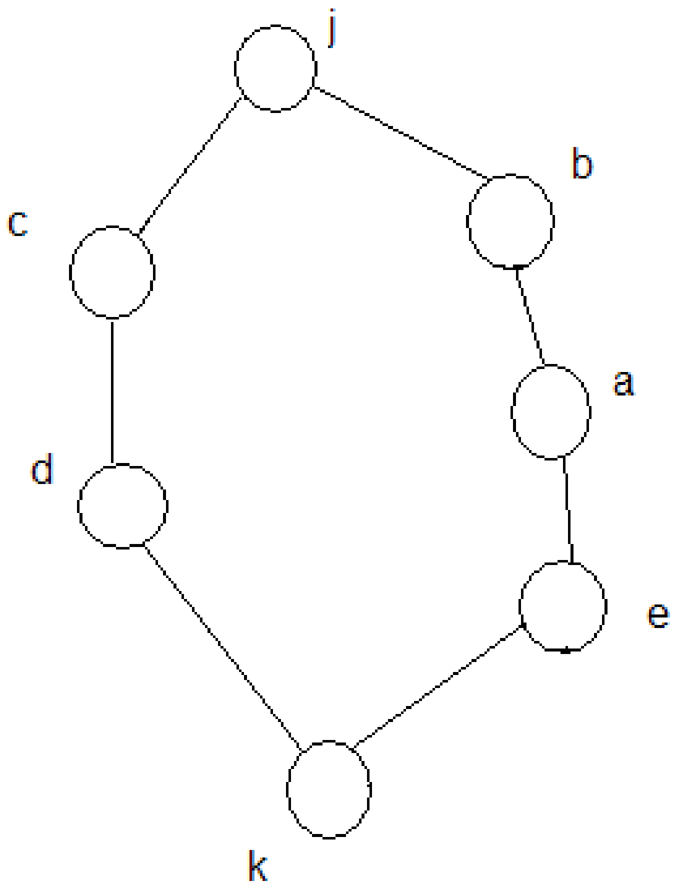


Figure 4.1: A network segment within a group.

node *k* will repeatedly get the update through *e* earlier than the one through *d*. Node *k* vacillates in its stand on the most current and best route to node *j*, thereby incurring a triggered broadcast every time its view changes.

To avoid such a situation, a node keeps track of the average inter-arrival time between the first routing update from a destination and the routing update containing the best path to the destination. When such a measure, called the *average settling time* for a route, is available, the advertisement of a route change is deferred for a time period equal to $2 \times Average\ Settling\ Time$

to allow for the arrival of the update carrying the best route. The incoming update is however accepted into the routing table and forwarding decisions are based on the update. If no routing update of a better metric is received in this time, the new route is immediately broadcast as it constitutes a significant change in the routing table at this node.

Aside from full dumps and triggered updates, incremental updates are broadcast periodically to disseminate information about stabilized routes. Incremental updates are used to convey information about the renewed availability of routes to certain destinations. Entries corresponding to destinations to which the Next_Hop and Metric fields have not changed fall in this category; SNs associated with these entries are expected to have changed. Since incremental updates are published at a more frequent rate, they have a limited size and may carry only a certain number of routing table entries. Then, the selection of the exact routing table entries to be advertised in an update should be done in a manner that will ensure fairness in the frequency at which each routing table entry is advertised. The fact that the number of routing advertisements possible in an incremental update is limited means that there will be an inevitable delay in the propagation of some routing advertisements: a table entry may be held back from advertisement in the current broadcast and instead advertised in the next broadcast. Therefore, a fair selection of routing table entries into incremental updates is necessary to ensure that each table entry receives a timely advertisement at least once in a while.

The lack of synchronization in the times at which broadcasts are made by different nodes results in a delay in the broadcast of routing information and in the suppression of the usefulness of such information. For instance, in Figure 1.1, assume that after the broadcast by node *j*, the phase relationship in the broadcast times of nodes *a, b, c, d, e* are $b < a < e < d < c$. Then, when node *j* broadcasts a routing update about itself with sequence number S1, the successive broadcasts by *b*, *a* and *e* propagate a routing update about j to *k*. *k* will then choose the route through *e* as its best (current) route to *j*. This is followed by a broadcast by node d, and a broadcast by node c intimating *d* of the route to *j* with sequence number S1. In the next update epoch, node *j* broadcasts a sequence number $S2 = S1 + 2$. This is broadcast by *b,a* and *e* so that *k* believes that the route through *e* with a sequence number S2 is the best route to *j*. Node d then broadcasts a route through itself to *j* of length 3 (which is shorter than the route through e of length 4) and sequence number S1. This update is however rejected by *k* since it is of a less sequence number than the route in its table. Node *k* will thus always think that the route through *e* is the best path to node *j*.

The sending of triggered updates nominally alleviates the problem by skewing the periodicity in the broadcast schedule of a node. This will hence re-order the phase relationships in broadcast times of a set of nodes. However, it is hard to determine how it will exactly take shape in a network: even if the relative broadcast times of a set of nodes is conducive to path establishment between 1 pair of nodes, it may hinder the propagation of routing updates between another pair.

The periodic exchange of routing information between group members ensures that every node in a group knows of all nodes which are also members of the same group as itself. Thus, when a new node joins a group, it uses its new GID towards the dissemination of routing information to other group members, who thereby learn of its affiliation to the group.

Disaffiliation of a node from a group is detected as follows. When a node decides to split away from a group, it secures a new GID for itself. Then, the routing updates sent by this node are ignored by any receiving node, and do not feature in the intra-group routing advertisements of the group. Discernment of this disaffiliation is implemented by a soft-state timer-based mechanism. When the route to a group member becomes invalidated, a timer, whose expiration time is approximately the time taken for a routing update to traverse the *Group Diameter*, is set off. If the route to the said node is still invalidated upon the expiration of the timer, the node is assumed to have left the group. The expiration time of this timer is determined as the time that it takes for a fresh route to the node to be made available given the network conditions. In the worst case, this time will be equal to $Group\ Diameter\ (in\ hops) \times Time\ Period\ of\ Periodic\ Broadcasts$. Therefore, when a route to a group member remains invalid for a certain period of time, the node is assumed to have disaffiliated itself from the group and its intra-group routing table entry is deleted; if a communicational requirement to this node arises, then the inter-group routing scheme will be employed.

4.1.1 Detection of interfaces to neighboring groups

When two groups come physically close to one another, then several nodes from each group enter within the communication range of nodes belonging to the other group. Such a node that serve as an interface for the communication between 2 different groups is termed as a *Border Node* (BN). A BN is not necessarily a node in the physical periphery of a group: if a node in the interior region of a group has a direct communication link to a node from another group, it is also a BN. The term 'Border Node' may be a misnomer in this sense.

When 2 groups come within the communication range of one another through the presence of a wireless link between a pair of nodes from each group, they are termed *neighboring groups*. A system for the detection of wireless links to nodes belonging to another group, and the dissemination of such information within the group is useful to establish routes between neighboring groups. Knowledge of routes to neighboring groups at each group in the network can pave the way for inter-group communication with any other group in the network.

As noted before, the intra-group routing mechanism requires the periodic broadcast of routing information by each node in a group. When a node hears a broadcast from a node with a different GID than itself, it discerns the presence of a communication link to a neighboring group. The node is then declared as a BN to the group from which it has heard the broadcast. A node may serve as a BN to more than 1 group, if it has a direct wireless link to a corresponding BN from each group. It is also noted that 2 groups may be interfaced by multiple BNs.

Just as with the mechanism to detect broken links between group members, when a BN does not hear of a broadcast message from its corresponding BN for a certain period of time it perceives a failure in the link to the BN. Even upon the failure of a link to a BN in a neighboring group, a node may continue to declare itself a BN to that group if it has a wireless link to at least one node in the neighboring group. In order to ascertain its status as a BN, each node therefore maintains a list called BN_NEIGHBOR_LIST which contains neighboring nodes which belong to other groups.

A node may therefore pronounce itself a BN to a neighboring group if it believes it has at least one wireless link to a node from that group. Each node maintains a table called the *BN table* which has the following fields, *Neighboring_Group*, *Border_Node* and *Sequence_Number*. Each entry in this table is interpreted as

"Node x is a BN to Group k. The last advertisement made by node *x* carried the Sequence Number(SN) n."

When a node identifies itself to be a BN to a group, it creates a BN table entry which designates itself as a BN to the particular group. Like with the intra-group routing mechanism, sequence numbers are used to denote the freshness of advertisements made about the BN statuses of nodes: higher the SN, more the credible the information is. When the table entry is initially created for the BN, an even SN is attached to it. When the link to a neighboring group fails (i.e., the BN has no more neighbors in that group), the node increments the SN of that entry by 1. An odd SN is used to denote a failed link status at the BN to the group specified in the entry.

When a node identifies itself as a BN to some group, it uses a base SN which it will associate with all BN table entries in which it is the Border_Node. This value is then incremented by 2 in these entries with each successive broadcast of BN table advertisements that it makes.

The SN associated with a BN table entry is generated only by the node in the Border_Node field of the entry, i.e., when a node believes it is a BN to a neighboring group it gives this entry an even number, and when the link fails, it invalidates the entry by supplying it with an odd SN. SNs have limited semantics in the case of BN advertisements. They are used to distinguish between active and failed links (even and odd SNs respectively). Also, a higher sequence-numbered update supersedes a BN table entry with a lower SN. So, when an odd sequence-numbered update about a BN is received, and the current BN table entry for that BN has an even SN which is of a lower value than that of the update, then the node accepts the update in the advertisement and invalidates the entry. When an even sequence-numbered update is received for a BN whose current entry in the BN table has a even SN of less value than the update, then the update is accepted, but this really changes nothing; it only so much as renews the BN status of Border_Node to Destination_Group. In other words, a BN does not need to repeatedly advertise its link status unless a change in status is detected.

So, when a node perceives a change in its status as a BN, that is, it has been newly interfaced to a neighboring group or has had its hitherto BN-ship disrupted, then it sends a BN advertisement. The BN advertisements are piggybacked on regular intra-group routing updates. When a change in BN status of a node to a group is detected it sends this advertisement in the next intra-group routing update. Thus, the sizing of intra-group routing updates may need to be modified if a BN advertisement is imminent.

A node that receives a BN advertisement necessarily accepts it if it has a higher SN than the corresponding entry in its BN table. If the node has no record of the advertised entry in its BN table, it accepts the advertisement if it has an even SN. When a node deems an incoming update fit for acceptance, it changes its current BN table entry and re-broadcasts this advertisement in its subsequent intra-group routing update. The BN advertisement propagates throughout the group in this manner.

It was mentioned before that it would suffice for nodes to advertise only changes in BN statuses, that is, a BN need not repeatedly advertise itself as an interface to a group. However, because of node mobility, it is not possible to guarantee complete dissemination of a BN update to all nodes in a group with the advertisement of only changes in BN statuses. For instance, if a node that has not heard of a particular BN advertisement moves into a region where the advertisement has already percolated, then it will not receive the update. To counter the effects of mobility, a small portion of the space in each intra-group routing update is reserved for BN advertisements.

When a BN table entry is invalidated it is deleted after a certain period of time. The deferral is to avoid the acceptance of an even sequence-numbered update of the same entry from a node that has not received news of the invalidation. The waiting time is approximately the time taken for the invalidation to reach all the nodes in the group.

The result of this process is that every node in the group knows of all neighboring groups in the network, and the BNs that interface to these groups. With this information, any node in a group can communicate with a neighboring group *X* as follows. From its BN table, the node sifts out a list of the candidate BNs to group *X*. From these, the closest BN *y* (the BN that has the best metric in intra-group routing table) is picked as the interface to group *X*. The packet is forwarded successively based on the intra-group routing table and reaches BN *y*. The BN *y* then rebroadcasts the packet to one of its neighbors in group *X*, a record of which is maintained in the list BN_NEIGHBOR_LIST as described before.

It is also noted that when a BN listens to a intra-group routing broadcast made by a node belonging to a neighboring group, it learns of the IDs of several members of the neighboring group. These Node ID-Group ID bindings are cached by the listening node to facilitate inter-group routing. More details on the caching of GID bindings are presented in the Section 4.2.4.

4.2 Inter-group Routing

The inter-group routing module serves to establish and maintain routes between any 2 nodes that belong to different groups. Group mobility induces a natural clustering in the network of nodes that belong to the same group. Such a network fragmentation allows the abbreviation of routes to all destinations in any other group in the network with the routing knowledge of simply one referential node in that group. The proactive intra-group routing mechanism complements the above route maintenance procedure by redirecting traffic away from the reference node towards the destination.

Then, with the given intra-group routing scheme in place, the functional requirements of the inter-group routing module may be rephrased as follows,

1. The inter-group routing scheme must be able to determine the current GID of the destination node, in the face of continual disintegrations and mergers between groups. So, a GID discovery scheme/service is required that can map the MANET address of a destination node to the ID of the group where it currently resides.

2. When a determination of the GID of the destination is available, the routing protocol must be able to supply the source node with a route to some reference node in the destination group.

The inter-group routing protocol is designed to be reactive in nature; that is, a route to a destination in a different group is determined only when a communication with the destination is sought at the source node. Before explaining why that is so, a recapitulation of the LANMAR protocol [12] is made here. LANMAR (Landmark Ad Hoc Routing) is a routing algorithm developed specifically for MANETs that contain groups. The intra-group routing scheme in LANMAR is proactive and uses an algorithm called Fisheye State Routing (FSR) to maintain routes to all nodes within the group. Inter-group routing is also performed in a proactive manner. Each group has a single node nominated as a leader, also called a *landmark* node. Distance Vectors (DVs) are propagated from all landmark Nodes to the entire network on a periodic basis so that all nodes in the network are aware of routes to all other groups in the network.

In LANMAR, it is assumed that although a source node knows the name of the destination it needs to communicate with, it can't determine the group to which the destination is bound to. That is, neither does a standalone service exist to furnish the GID of a named node in the network, nor does the node address assignment mechanism incorporate the GID of the node's affiliation into its regular MANET address. Therefore, when a source node desires to communicate with a destination, it first queries all the groups in the network to ascertain the GID of the destination. An acknowledgment is sent from the group that contains the destination to the source, following which communication begins.

The inter-group routing module is developed over the same assumptions on GID ascertainment as the LANMAR protocol, wherein the GID affiliation of a destination is not static and there exist no separate services for this information to be made available to the source. A reactive routing mechanism that encapsulates GID determination is chosen on the rationale that a reactive querying procedure is required anyway to consummate route establishment in the proactive inter-group route maintenance scheme of LANMAR. It is noted that the periodic exchange of group memberships between different groups, piggybacked on intra-group routing updates, is infeasible because the sheer volume of the data makes the associated costs prohibitive.

A second reason why a periodic broadcast-based inter-group routing scheme is not employed is that the the skewed nature of the scheduled routing broadcasts by nodes results in considerable delay in the propagation of a routing update to distant nodes in the network. In LANMAR, DVs from each landmark node are broadcast to all nodes in the network periodically. The exact fashion in which a node receiving a DV from a landmark re-broadcasts it has not been described in [12]. 2 possibilities are considered hereon: an instantaneous re-broadcast by any node receiving a DV, or, the incorporation of the DV into the routing update in the scheduled intra-group advertisement. The immediate re-broadcast of DVs by every node in the network is very expensive when considering the number of landmark nodes in the network and the frequency at which such an update must be made to sustain route maintenance. Therefore, network-wide flooding of DVs with immediate propagation by forwarders is ruled out.

When the successive propagation of DVs is embedded in the scheduled intra-group routing broadcasts made by intermediate forwarders, the staggered timing of these routing broadcasts means that an update from a landmark node may reach a node *N* hops away about $N \times X$ seconds later (worst-case), where *X* is the time interval between 2 routing broadcasts made by a node. Thus, an update received at a node may be as old as $N \times X$ seconds in the worst-case. Reparation of route failures especially from the movement of the destination group will have to wait until a routing update from the destination group reaches the source which may incur a long delay from the non-immediacy in the propagation of routing updates.

All reactive routing protocols require the immediate propagation of Route Request (RREQ) messages by receiving nodes (e.g., [1, 8]). The control overhead generated in the network-wide flooding of instantaneously rebroadcast messages in an on-demand scheme is proportional to the number of connection pairs that the network must support; so, when the number of connection pairs is large, reactive inter-group routing also incurs considerable overhead. A forwarding policy on RREQ messages that capitalizes on the network structure and the already in-place intra-group routing scheme to curtail the flooding of RREQs to a subset of the network nodes is described next.

When a source node can not find a route to a destination in its intra-group routing table, it infers the affiliation of the destination to another group in the network. The inter-group routing module is then invoked to discover a route to the destination. The source creates a RREQ packet that it broadcasts to the network so that the destination hears of the RREQ query and replies to it.

As explained before, a single reference node in a group may summarize the routes to all its group members in its routing update in a proactive inter-group routing scheme. Analogously, in on-demand inter-group routing, a RREQ query need only reach a single reference node in every group of the network for an accurate response to be made. The reference node in each group, based on its intra-group routing table, can determine if the sought destination belongs to its own group and hence appropriately respond to the query. Therefore, network-wide flooding of the RREQ message is not required. Instead, contained flooding of the RREQ packet to a subset of the network nodes is accomplished by an overlay on the intra-group routing and BN dissemination schemes.

An abstracted network graph in which each group is a vertex is considered hereon. An edge exists between vertices *a* and *b*, if 2 coordinate BNs, one each from the groups represented by *a* and *b*, share a wireless link. Vertices *a* and *b* are said to be neighbors and the groups they represent are neighboring groups.

A source node can propagate a RREQ message to all its neighboring groups from its intra-group routing and BN tables as follows. The source node knows of all its neighboring groups and the BNs that interface to each group from its BN table. For each neighboring group, the source sifts all candidate BNs for the best metric (shortest in hops) in its intra-group routing table. A determination of the next hop to the closest BN to each neighboring group is thereby made. An RREQ is multicasted to each next hop determined from the above process. The receiving nodes forward the RREQ so that it reaches one and only one interface to every neighboring group. Each

BN that receives the RREQ packet then identifies its coordinate BN in the neighboring group and passes on the RREQ to it. The RREQ is thus propagated to one node in every neighboring group of the source node.

The receiving nodes (BNs) attempt to retrieve a route to the destination from their own knowledge of the network topology. If a route is unknown, the node must forward the RREQ to other groups that neighbor its own group. To this end the node multicasts the RREQ packet to all its neighboring groups (except the group from where the RREQ last egressed) in the same manner as the source of the RREQ packet. The repeated propagation of the RREQ ensures complete coverage of all groups in the network by the RREQ packet, subject to the timely availability of information about neighboring groups.

In order to distinguish between successive RREQs sent between the same source-destination pair, a *Sequence Number* (SN) is associated with every RREQ sent by a source. SNs are, as usual, used to denote the freshness of a RREQ, and are generated by the source of the RREQ. With every successive RREQ generated by the source, the associated SN is incremented. Whenever a node receives a RREQ packet, it keeps a record of the source-destination pair in the RREQ and the associated SN. If an RREQ between the same source-destination pair is subsequently received of the same SN (through another path of propagation), it is declared a duplicate and hence discarded. Thus, SNs provide for the detection and elimination of redundant RREQ threads in the system.

The RREQ packet also contains a field called the *Traversed_Path* which contains a list of all groups already visited by the RREQ thread. The source first includes its own GID into the list when it initiates the RREQ broadcast. Then, upon the ingress of the RREQ packet into a new group, the receiving node appends its own GID into the list. One of the reasons why such a list is made is to avoid the rebroadcast of the RREQ packet to groups already featured in this list. When an intermediate node is to re-propagate the RREQ packet to its neighboring groups, it excludes the groups that have already been traversed, as listed in the field Traversed_Path.

Despite the mechanisms used to detect redundant RREQ's in the system, the lack of coordination between different BNs through which different RREQ threads ingress a group, results in the participation of more nodes in the RREQ flooding than necessary. This is exemplified next.

Consider the network topology in Figure 4.2. There exist 5 groups in the network - *A, B, C, D* and *E*. All the nodes marked in the figure with thick black dots are BNs and the lines joining BNs denotes the link between the different groups that they interface. For instance nodes *5* and *9* interface groups *A* and *B*. Let a source node *45* in group *A* query for a route to destination *100* in group *E*. Groups *A, B, C* and *D* are all mutual neighbors and are interfaced by 4 distinct pairs of BNs such that no node is a BN to more than 1 group. Group *E* is connected to groups *C* and *D* through 4 different pairs of BNs each, again with no BN serving as an interface to more than 1 group.

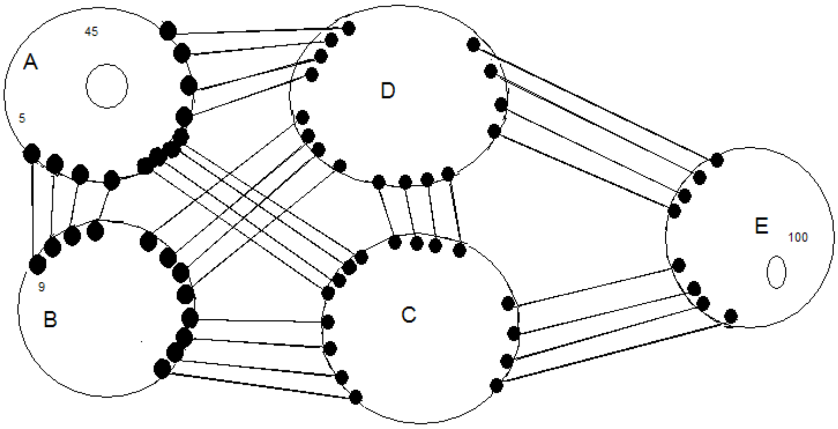


Figure 4.2: A network scenario used to illustrate RREQ threads in inter-group routing.

The number of RREQ threads propagated through the network is analyzed based on the RREQ forwarding scheme described before. In this scheme, when an RREQ egresses a group, the counterpart BN that receives the RREQ sends it to all its neighboring groups (except the ones listed in the Traversed Path field). Also, any node that has heard of the RREQ before simply rejects it. In the worst case, when the paths used by all BNs in forwarding to neighboring groups are disjoint, then, the number of RREQ threads in the network is equal to the number of paths (denominated over groups and containing no loops. E.g.,:A→C→E is a path) from the source group that are possible in the network. Thus, in the above example scenario, in the worst case there are 12 such RREQ threads as listed below:

1. A→B→C→D→E

2. A→B→C→E

3. A→B→D→C→E

4. A→B→D→E

5. A→C→D→B

6. A→C→D→E

7. A→C→B→D→E

8. A→C→E

9. A→D→C→E

10. A→D→C→B

11. A→D→B→C→E

12. A→D→E

Only considering threads 3 and 7, it is noted that although the RREQ packet that reached group *B* was retransmitted to group *D* in thread 3, the RREQ from Group *C* is accepted and retransmitted to group *D* again in thread 7. The redundant re-propagation of the RREQ in thread to group *D* happens because there is no coordination between the relevant BNs to discard duplicates (assuming the 2 threads use different paths). Therefore, more nodes are engaged in the RREQ propagation process than necessary. The above set of threads forms the worst-case behavior of the network. If however, some of these threads required the participation of the same node more than once, then they could be rejected when they are heard of for the second time.

In order to provide for a systematic coordination mechanism to detect and discard redundant RREQ packets, an RREQ packet in ingression into a group is detoured from the receiving BN to the GL of the group. The GL acts as a *collector* of all equivalent RREQ messages and re-propagates only one copy to its neighboring groups (excluding groups listed in Traversed_Path).

The savings from the new forwarding scheme are analyzed over the same network scenario in Figure 4.2, when a RREQ is sent from *45* in group *A* seeking destination *100* in group *E*. The RREQ threads that evolve in the new forwarding scheme that requires a digression of the RREQ packet to the GL are enumerated in Table 4.1.

At Stage#1, the RREQ originates at group *A* and is sent to groups *B, C* and *D*. The receiving BNs in these groups send the query over to their respective GLs which decide the other groups to which the packet must be sent to. Groups already visited by the packet are avoided. So, groups *B, C* and *D* send the packet to all their neighbors except group *A* which has already been visited by the packet. Thus at Stage#2, *B* sends the RREQ to groups *E,C* and *D*; *C* to *E,D* and *B*; *D* to *C* and *B*. The receiving BNs in group *E* will send an RREP to the RREQ. All the other RREQ threads received at the other groups will be forwarded to the GL and then rejected.

Table 4.1: Evolution of RREQ threads in the new forwarding scheme.

Stage#1	Stage#2	Stage#3	Stage#4
A	B	C	Reject
		D	Reject
	C	E	Route Reply
		D	Reject
		B	Reject
	D	C	Reject
		B	Reject
		E	Route Reply

The new forwarding policy thus incurs the participation of fewer nodes (8 threads with maximum path length of 3 versus the 12 threads with a maximum path length of 5 in the former scheme) in a worst-case comparison between the 2 schemes.

For a guaranteed coverage of all groups in the network by the RREQ packet, a RREQ packet that is re-propagated by the GL may not be discarded by forwarders during the egress of the packet from the group, irrespective of whether it has been heard of before or not. This constraint introduces a new field into the RREQ packet, *Reject*. A value *0* denotes that the RREQ packet may be discarded by forwarders if it is a duplicate copy, and a value *1* means that the packet must be forwarded, as dictated in the RREQ contents, even if an equivalent RREQ has been heard before. An RREQ packet that has just entered a group is stamped a *0* value in its Reject field by the receiving BN. It is then forwarded to the GL. When the RREQ thread is reprised by the GL, the Reject field value is made 1 so that it is necessarily delivered to all neighboring groups. The RREQ packet contains the following fields:

1. Source_Address.

2. Source_Group_ID.

3. Destination_Address.

4. Destination_Group_ ID (if available).

5. Sequence_Number.

6. Reject (Binary value).

7. Traversed_Path (Of groups).

8. Next_Group(s) (Array of Groups).

9. Next_Hop(s) (Array of nodes).

The source of the RREQ packet fills in its own address and GID in fields 1 and 2, and the destination node's address in field 3. If the destination's GID is known, it is entered into field 4. All nodes maintain 2 types of caches: a *Route Cache* and a *GID Cache*. The GID cache contains 2 fields: *Node Address* and *Group ID*. Whenever a node hears of any bindings in the control packets exchanged in the inter-group routing procedure, it adds the information to its GID Cache.

Also, during the exchange of control packets (such as a RREQ packet), nodes extract information about any routes made known in the packet and add it to their Route Caches. The routes published in the control packets are all denominated in terms of *groups*. The information maintained in the caches is purged after a certain time. Also, from the dissemination of BN advertisements, all nodes in a group learn of their neighboring groups and add routes to these groups to their Route Caches (again denominated over *groups*). The routes to neighboring groups are not subject to elimination from the expiry of timeouts; the explicit invalidation advertisements made by BNs are used to purge failed route entries falling in this category.

A source node that seeks an inter-group communication, first attempts to determine a route to the destination from its GID and Route Caches. If it does not possess a route, only then does it initiate a route request procedure.

Field 5 is filled in with an incremental SN for every successive route request made by the source. With every successive RREQ initiation, the source increments the SN by 2. As stated before, SN is used to detect duplicate copies of a RREQ packet. When a node hears a RREQ packet for a particular source-destination pair with the same SN as before, it infers that it is a duplicate copy. The Reject field is used to dictate if a current RREQ packet may be discarded when it is a duplicate of a previously heard RREQ. A *0* value in this field means that the packet may be discarded if it is a duplicate, whereas the packet must be processed if the field has a value *1*. As noted before, a RREQ packet egressing a group may not be discarded even if another copy of it has been heard before at a node. So, when the source initially broadcasts the RREQ, the Reject field is set to be *1*. (The field carries no context in the source's group since no other node could have possibly heard the RREQ before).

The Traversed_Path is a list of all groups that have been currently traversed by a particular RREQ thread. The source stamps its own GID into this field. The source of the RREQ packet adds the list of all its neighboring groups to field 8. This is a list of all groups that the RREQ must reach next. Also, the source sifts through all the candidate BNs to each neighboring group for the closest BN to each neighboring group. The intra-group routing table field Metric is used to evaluate the best of the candidate BNs to each neighboring group, and the corresponding Next_Hop to the *best BN* is looked up. Thus, the source knows of all the neighboring groups the RREQ must reach and the corresponding next hops the RREQ must pass through to reach the neighboring groups. The list of neighboring groups is added to field 8, and the respective next hops are added to field 9. The packet is then broadcasted.

When a node receives a RREQ packet, it inspects the field 9 to see if it is an intended recipient of the transmission; if it is not it summarily discards the packet. If it is an intended recipient of the RREQ packet, it checks to see if it has already processed the particular RREQ before. If it has a record of processing the RREQ packet before, and the Reject field is set as *0*, then the packet is rejected. Otherwise, the details of this instance of the RREQ (i.e., source-destination pair, SN) are made note of. Next, the node processes the query to determine if it possesses a route to the destination. To this end it looks for a route to to the destination in its intra-group routing table, failing which, it attempts to extract the GID of the destination (from its GID cache or from the RREQ packet, if available), and a route to the destination's group from its Route Cache. If a route is known or the node that receives the query is the sought destination, then a Route Reply (RREP) is made to the source. Otherwise, the node forwards the RREQ packet as follows.

If the packet is still on the egress path out of the group, the node determines all the groups to which it is responsible for forwarding the RREQ. This consists of all the groups in field 8 to which current node is listed as the next hop listed in field 9. A determination of the next hop on the shortest path to the each of these groups is made from the intra-group routing and BN tables. The node then reconstructs field 8 in the RREQ packet to include only those groups to which it is responsible for forwarding the RREQ. The corresponding next hops are filled in Field 9 and the packet is broadcasted.

If the packet has however reached a group that it was meant to, then the first node in the group to receive the packet stamps its GID in the Traversed_Path field and sets the Reject field value to *0*. This means that the RREQ packet may be discarded in subsequent transits if it is a duplicate. Now on, the packet must be transferred to the GL, which will decide on its re-propagation. So, if the node receiving the packet is not a GL, it determines the next hop to its GL from its intra-group routing table and sets this as the value in field 9. Field 8 automatically remains the GID of the current group that the RREQ has reached. If the RREQ packet has reached the GL, the GL determines all its neighboring groups that have not been featured in the Traversed_Path field of the RREQ. These are the groups to which the RREQ must be subsequently forwarded. The field 8 in the RREQ packet is filled in with the GIDs of these groups, and the corresponding next hops on the shortest path to each group are entered into Field 9. Also, the Reject field of the RREQ packet is made *1* so that no subsequent recipients within the group discard the packet. The RREQ packet is then rebroadcasted.

In this fashion, the RREQ reaches all groups in the network. Network-wide coverage of the route request can be guaranteed if the BN advertisements are made in a timely manner. It is noted that all intermediate forwarders of RREQ packets cache the GID bindings of the source and destination (if available), as well as the route to the source of the RREQ from the Traversed_Path field. As noted before, all routes are denominated in terms of groups. A route is thus a loop-free sequence of GIDs which may be used in conjunction with intra-group routing and BN tables to set up a path between 2 nodes belonging to different groups.

4.2.1 Route Reply Generation and Forwarding

When the route request has reached either the destination to which a route is sought or an intermediate node that possesses a route to the destination, a Route Reply packet (RREP) that contains the route from the source to the destination is generated and forwarded to the source of the RREQ. Each intermediate forwarder inspects its intra-group routing table, GID Cache and Route Cache for a route to the destination sought in the RREQ. If the destination belongs to the same group as the recipient of the RREQ, then a route to the destination must be available in the recipient's intra-group routing table. If not, the node attempts to extract a GID binding for the destination from its GID cache (or from the RREQ packet if it is published there). When a GID binding for the destination is available, the Route Cache is inspected for a route to the destination's group. When more than one such route is available, the shortest route (fewest hops in terms of groups) is chosen as a reply in the RREP packet. In most circumstances, a RREP is generated by the BN of the group containing the destination, which is the point of entry for the RREQ packet into the destination's group. The generated RREP packet is to contain the following fields:

1. Source_RREP.
2. Source's_GID.
3. Route_Requester.
4. Requester's_GID.
5. Forward_Route.
6. Reply_Route.
7. Sought_Destination.
8. Destination's_GID.
9. Next_Hop.

The source of the RREP adds its own address and GID to Fields 1 and 2, the address and GID of the solicitor of the route to Fields 3 and 4, and the address and GID of the sought destination (in the RREQ) to fields 7 and 8 respectively. Field 5 (Forward_Route) contains the route that the RREP must use to reach the source of the RREQ. This route is the reverse of the field Traversed_Path in the RREQ packet. Field 6 (Reply_Route) is the actual route to be used by the route solicitor to communicate with the destination. When the replying node belongs to the same group as the destination, this is the same as the Traversed_Path field in the RREQ. If however, the respondent is a node belonging to a different group, then field 6 is the loop-free concatenation of Traversed_Path and a route from the respondent to the destination's group.

The forwarding of the RREP is done based on Forward_Route that the RREP must use to reach the route solicitor. Every forwarder determines the next group to which the RREP must be forwarded from Forward_Route. Then, as usual, the next hop on the shortest path to this neighboring group is determined and filled in field 9. The RREP is then sent over to the next hop.

All intermediate forwarders of the RREP packet cache the GID bindings of the addresses in fields 1,3 and 7. Further, all nodes also cache the routes to the group from which the RREP emanated, and the group where the RREP will terminate. When the RREP reaches the node that had originally queried the network for a route, all forementioned GID bindings as well as the route to the destination's group are cached. Then, all data packets queued for transfer to the destination to which a route has been made available are sent away.

More than 1 RREP may be generated in the network because the RREQ packet radiates from the source node into several threads. No attempt to curtail the forwarding of redundant RREPs (through the use of SNs, for an example) is made because variable propagation delays in different network segments can delay the arrival of the *best route* to the destination to a time beyond the arrival of the first RREP about the destination.

4.2.2 Data Packet Forwarding and Source Routing

When the source node is aware of a route to the destination, it transmits all data packets that are queued for the destination. Data packets carry the inter-group route that is to be used for packet forwarding by all intermediate nodes. The source node selects the shortest route (assessed on the number of hops over groups) to the destination's group and appends this route to the data packet.

If the discrimination of the shortest route is to be based on the actual number of wireless hops between the source and the destination, then information on the number of hops traversed by the RREP packet must be made available. All forwarders inspect the route dictated in the data packet for the ID of the subsequent group the packet must be sent to. Then the next hop on the shortest path to that group is determined from the intra-group routing and BN tables, and the packet is propagated to this node. When the data packet reaches the destination's group, all forwarders use their intra-gropu routing tables to determine the next hop to the destination.

Termed *source routing* [8], the appendage of the route that the data packet must adhere to provides for more resilient packet forwarding in the event of intra-group node mobility. The source route is a loop-free sequence of GIDs from the source of the data packet to its destination that provides direction for packet forwarding to intermediate receivers. Even when the topology of a group changes, any node that receives an inter-group communication (data packet) that is to be forwarded has to only lookup the source route to determine the next group to which the packet must be forwarded.

If data packets did not carry source routes, then only the actual participants in the forwarding of the RREQ and RREP packets between a source-destination pair would know of the associated routes. Then, in the event of intra-group node mobility, a node that has not participated in the route establishment phase may receive a data packet and not know where to forward the data packet resulting in the termination of packet forwarding.

Thus, the provision for source routing in the data packet makes the inter-group packet forwarding process more resilient to intra-group topological changes. Of course, when the topological changes are such that a node receiving a data packet does not possess a route to the BN interfacing the *next group* dictated in the source route, then the packet must be queued until the intra-group routing module remedies the situation.

It is thus noted that different data packets transferred between a source-destination pair may use different paths (denominated in terms of nodes) even when they adhere to the same source route (denominated over groups).

4.2.3 Route Error (RERR) Notification

A Route Error (RERR) message is generated in reactive protocols like AODV, operating on a conventional MANET environment, when a forwarder of a data packet discerns that the next hop on the forwarding route is inaccessible. A RERR notification is sent back to the source of the communication and route re-establishment is initiated by the source. The inter-group routing mechanism in this work, however, need only worry about route failures that emanate from the disruption of links between 2 neighboring groups. When the intra-group segment of an inter-group communication path fails, the onus of recovery is upon the intra-group routing mechanism.

In the abstracted network graph that contains only groups, an edge between 2 neighboring groups is broken only when the links between all neighboring coordinate BNs in the 2 groups are broken. Every node in a group maintains its neighborhood topology on this graph through an aggregated view of the link statuses advertised by the BNs of its group. In order for a node to deduce a route failure to a neighboring group, it must receive an invalidation update from all the BNs that had previously advertised themselves as interfaces to that group. When all links to a neighboring group are disrupted, various group members infer the route failure at different time points. Therefore, a route failure in an inter-group communication may be detected by any node of the group that lies on the communication path.

When a data packet forwarder determines that the next specified group in the source route that the packet must reach is inaccessible, it drops the data packet and prepares to notify the source of the route failure. A second kind of route failure happens when the destination splits away from its group half-way through an inter-group communication. In this case, a packet forwarder in the group to which the destination was hitherto affiliated observes the abandonment from the missing entry for the destination in its intra-group routing table. This node must then send a RERR to the source indicating the discrepancy in the destination's GID binding.

The RERR packet is of the following format:

1. Source_RERR.

2. Source's_GID.

3. Destination_RERR.

4. Destination's_GID.

5. Forward_Route.

6. Failed_Node.

7. Failed_Route.

8. Failed_GID_Binding.

Fields 1 and 2 contain the address and GID of the source of the RERR, while Fields 3 and 4 are the address and GID of the source of the data packet whose route has failed. Field 5 contains the route that must be used to forward the RERR packet. This is simply the reverse of the segment of the source route in the data packet between the source group and the group that generates the RERR. Field 6 is the address of the destination of the data packet, that is, it is the node to which the route has failed. Field 7 is filled in when the RERR is generated because two neighboring groups have had their link severed. This entry is a route consisting of the GIDs of the 2 said groups. Field

8 is filled in when the RERR notification is made because the GID binding of the destination was erroneous in the data packet. This *failed binding* of the destination is entered into Field 8.

The RERR packet is forward by nodes based on the route specified in the RERR packet. All forwarders cache all extractable GID bindings and routes from the RERR packet. Forwarders also use the Fields 7 and 8 to invalidate any entries in their GID and Route Caches respectively.

As noted before, when a route failure when 2 groups occurs, different nodes on the communication path detect this failure at different points of time. The belated detection of route failure at some node in the communication path allows a data packet to penetrate deep into a group, without being dropped. When a data stream is being transferred from a source to a destination, different packets belonging to the stream are buffered at different nodes at the same point of time. The combination of the 2 conditions can result in the generation of RERR messages by every node on the communication path that learns that the subsequent group in the source route is unreachable. To avoid the generation of superfluous RERRs, every forwarder of a RERR makes note of its occurrence. Specifically, the addresses of the source-destination pair are cached for a certain time period, and all equivalent RERRs generated within this time are discarded.

When the source of the failed inter-group communication receives the RERR, it invalidates the entries in its GID and Route Cache. Then, it initiates a RREQ as part of the route re-establishment procedure. It is possible that receivers of this RREQ supply a route reply that contradicts the *failed information* presented in the previously received RERR message. To nullify this possibility, RREQ packets contain 2 additional fields

10. Failed_Route.

11. Failed_GID_Binding.

Accordingly, the source of the RREQ copies the information in Fields 7 and 8 of the RERR to the Fields 10 and 11 of the RREQ packet. This ensures that any receiver of the RREQ considers the *failed information* presented in the RREQ before it deems a route worthy of a RREP.

4.2.4 Setting Cache Timeout values

All routes and GID bindings known from the broadcast of the control packets (RREQ, RREP and RERR) as well as data packets are cached by intermediate forwarders. Also, nodes that overhear packets exchanged between other nodes may cache such information. Since there is no indicator of the time a route will remain active or the duration for which a node will remain affiliated to a group, the timeout values for the Route Cache and GID cache entries are set by experimentation. Large timeout values result in the prolonged caching of potentially spurious data, and small timeout values render the caching process sub-optimal.

The timeout value for the cache that records the reception of a particular instance of a RREQ message at each node is set as the maximum anticipated time for which the RREQ will float around in the network before all the RREQ threads are either serviced or discarded. This is approximately the time elapsed in the propagation of a RREQ along the network diameter.

Also, a fourth type of cache is used to retain information on the notification of a route error on a inter-group communication path. This information is cached by all nodes that forward the RERR message to the source. The timeout value for this cache entry is approximately the time elapsed before the RERR packet reaches the source and a route re-establishment procedure is triggered. Additional details on choosing cache timeouts and caching strategies for on-demand protocols can be found in [47, 48].

AODV uses additional optimization techniques like *expanded ring-search* and route salvage at an intermediate point of failure [1]. These have not been attempted in the inter-group routing scheme. Also, details on on-demand routing such as the generation of SNs, Time-to-live (TTL) restrictions of RREQ messages etc. can be found in [1,8].

The routing protocol proposed in this work is briefly described in Appendix A as a collection of event-driven procedures operating at the routing layer of each mobile node in the MANET.

4.2.5 Evaluation of the Inter-group Routing Protocol

The inter-group routing scheme capitalizes on the inherent clustering in MANETs that contain groups to limit the flooding of the RREQ packet to a subset of the network's nodes. Reactive protocols are generally unsuited to environments with heavy traffic needs because of the repeated network-wide flooding of RREQ messages towards route establishment. AODV generates more control overhead from broadcasting RREQs than the amount of data transferred, when the number of connection pairs is large. The controlled flooding technique of the inter-group scheme requires the participation of only a select number of nodes in the network, and may therefore not incur so much of an overhead under heavy loads. Overall, the transmission of RREQs along select paths through the network is similar to the *inter-zone* routing scheme of ZRP [24].

It was noted earlier that groups, being specialized sub-networks in the MANET, may desire autonomy in the choice of their intra-group routing scheme depending on their sizes and traffic requirements. However, the proactive discovery and dissemination of BN information is an underpinning to the structure through which RREQs flow into and out of each group in the network. Thus, the overall design of the routing protocol rules out the use of a reactive intra-group routing scheme.

The BN discovery and exchange protocol makes explicit advertisement of all BNs that interface a group with a neighboring group. The knowledge of multiple BNs to a neighboring group provides for more resilient inter-group routing. Even when the link between a BN and its neighboring group is down or a route to a BN is invalidated, a forwarder may still be able to route a data packet from the knowledge of other BNs that interface the pertinent neighboring group.

An important requirement of the protocol is that its performance may not degenerate significantly when a large number of *individual nodes* are present in the network. Individual nodes are themselves treated as groups by the protocol: so communication originating and/or terminating at an individual node is pursued by the inter-group routing module.

In LANMAR, a number of approaches to handle individual nodes have been speculated. Having all individual nodes broadcast their DV updates to the rest of the network, similar to the inter-group route maintenance by landmark nodes, has been ruled out as being expensive. Another approach proposed is to use reactive routing when a route to an individual node is sought. This requires the source to first conclude that the destination is an individual node (by querying all groups in the network) followed by a network-wide broadcast of the RREQ. Routing to/from individual nodes is naturally subsumed into the inter-group routing mechanism in this work. The broadcast of RREQ packets is limited to a fewer number of nodes (depending on the percentage of individual nodes in the network) compared against the network-wide flooding required for reactive routing to individual nodes by LANMAR.

Being reactive, the inter-group routing protocol suffers from a route acquisition delay, which is the time elapsed between the generation of the RREQ and the reception of a RREP at the source. In LANMAR, routes are continually maintained to other groups in the network. A route acquisition delay is incurred whenever the GID binding of a destination is unknown, in which case a follow-up querying procedure is initiated to determine the same.

The fact that RREQ propagation is made only on selected paths through the network translates into the use of sub-optimal communication paths by the inter-group routing module of this work: there is no guarantee that a RREQ packet will reach a destination on the shortest path from the source. The detour of the RREQ packet to the GL at each group elongates the route further, making the customary reverse path set-up for RREP propagation (by forwarding the RREP to the sender of the RREQ, such as in AODV) injudicious. RREPs may therefore only carry routes that are denominated group-wise. With only group-wise routes available, intermediate forwarders of data packets may discriminate the best path to reach the next group specified in the source route, using their intra-group routing and BN tables. The aggregated intra-group segments may form a sub-optimal inter-group communication path as shown in Figure 4.3. The source and the subsequent forwarders choose the path in *red* (single solid line) over the shorter path in *blue* (two thin parallel lines) because only an abstracted view of the route between the source group and destination group is available. Nevertheless, the group-wise denominations of source routes offers more immunity to intra-group topological changes as explained before.

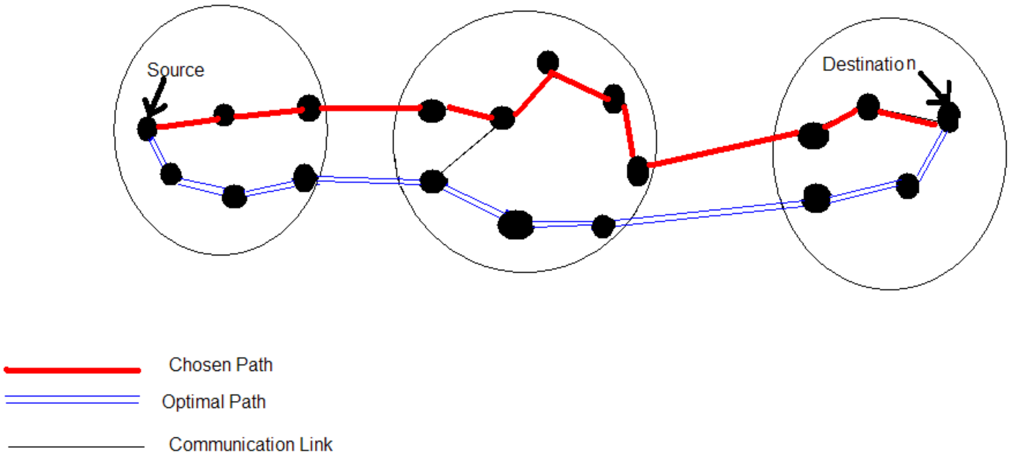


Figure 4.3: Illustrating the choice of sub-optimal paths by the inter-group routing scheme.

The indirection of each RREQ packet to the GL may create a performance bottleneck inside the wireless range of the GL. The purging of redundant RREQ packets, which were previously received or overheard, helps alleviate the situation.

In summary, LANMAR and the protocol in this work extend proactive and reactive routing, in conventional ad hoc networks to MANETs that contain groups, respectively. LANMAR exploits the grouped structure in the network to reduce the volume of each inter-group routing update by a factor of the average group size. That is, each routing update in the network carries only the intra-group routing update (equal to the size of the group) and the updates sent by the landmark nodes. Reactive routing in this work controls the extent of flooding of RREQ messages over a framework constructed from the exchange of BN advertisements. The downside of a proactive periodic flooding-based inter-group routing scheme is that the routing updates may reach distant nodes after a long delay due to the lack of synchronization in the broadcast times of intermediate forwarders. Reactive routing, on the contrary, is usually performed by the immediate re-broadcast of RREQ packets by forwarders; having said that, the repeated flooding of RREQ packets from heavy connection requirements or frequent route failures can potentially overwhelm the network.

It is also noted that the inter-group routing scheme proposed herein is pseudo-reactive, for route maintenance to neighboring groups is performed in a proactive manner by the BN dissemination strategy. The timely advertisement of routes to neighboring groups is crucial towards the performance of reactive route solicitation using RREQ packets, and broadly, for all inter-group communications. For instance, if the BN advertisement mechanism failed to intimate a source node of the availability of a route to a neighboring group, it results in the exclusion of the group from the list of immediate recipients of the RREQ packet sent by the source.

CHAPTER FIVE
PERFORMANCE EVALUATION BY SIMULATIONS

5.1 Simulation Environment

The performance evaluation of the protocol presented in this work is carried out by simulations of the protocol's operation on various types of MANETs in Network Simulator-2 (ns-2) [49]. Ns-2 is a network simulation tool that provides interfaces to the creation and simulation of networks (wired and wireless) according to a variety of protocols at different layers of the network stack. Integration of the protocol in this work into ns-2 involves specifying the routing methodology that nodes will adopt to forward packets by utilizing the standard interfaces in ns-2 that model various network components and functionalities. The implementation of the routing protocol is done in C++.

Ns-2 may be configured to generate traces of different kinds of events that accompany the simulation of a network scenario (e.g., transmission of packets, dropping of packets at interface queues, movement of nodes, reception of packets at different layers). The trace files provide data whose statistical analysis may yield different performance attributes of the simulated protocols (e.g., throughput, packet delay, volume of control overhead).

A comparison of the routing protocol in this work is also made against AODV [1] and DSDV [2], whose implementations are part of recent ns-2 releases (e.g., ns-2.33). In the following, the routing protocol in this work is denoted as GMRP (Group Mobility Routing Protocol) for an easy reference. The performances of GMRP, AODV and DSDV on MANETs (that exhibit group mobility in a predominant number of cases) are compared in terms of the *network throughput* and *average packet delay* of received packets. Network throughput is calculated as the sum of the throughputs of all network connections. The throughput of a connection is the number of packets received by the destination divided by the time duration of the connection. In the context of the routing layer, the network throughput is a measure of the ability of the routing protocol to route and deliver packets between connection pairs. The second measurement sought for comparison of the protocols is the average packet delay. This is the average value of the delay suffered by all packets in their transit from the source to the destination. It is noted that only the routing methodology and not its implementation has a bearing on the packet delay (i.e., the number of lines of code, the style of coding etc. do not affect this value). With respect to the routing layer, this value is reflective of how prompt the routing protocol acquires routes toward a desired connection, the level of packet queuing that is implemented and the length of the paths used for communication. A high throughput and low packet delay are plausibly compatible and interdependent objectives; a separate analysis of each is still made owing to the large number of protocol characteristics that factor into their computation. The delay suffered by packets that have not been delivered to their desired destinations has been discounted in the computation of the average packet delay.

5.1.1 Modeling of Node Mobility

The experimentation of the protocols' performance involves the creation of appropriate network scenarios on which the protocols are simulated. The network scenarios were created based on the following mobility model. The motion of groups in the network is independent of one another. The network is thought of as containing a certain number of standard checkpoints, to one of which every group in the network targets its motion towards. Thus, each group chooses one checkpoint in the network and starts moving towards it. Once the checkpoint is attained, a different checkpoint is chosen and motion continues onwards. This process is repeated throughout the simulation duration by every group in the network. Nodes within a group are initially positioned such that they form a connected network within themselves. This consists of defining an appropriate portion of the network area, in which the group members will initially reside, termed the *group area*. The group area is a square and is divided into a number of square grids, at the corners of which a group member may be positioned. The length of the sides of the group area and the square grid depend on the number of nodes belonging to the group as well as the *average number of neighbors per node* desired. When the group begins motion towards a checkpoint, the group members randomly choose a new grid corner in the group area, formed about the target checkpoint, to move to. Thus, the spatial arrangement of group members with respect to one another changes with movement from one checkpoint to another, but, group members always form a connected network amongst themselves. It is noted that some randomness in the motion parameters of nodes is induced which may lead to the temporary loss of connectivity within the group. Overall, the model emulates the spatial proximity of nodes moving as a group, and also incorporates a reasonable rate of changes in the network topology of the group. It is similar to the RPGM model [16] for group mobility, except that the topology of the group is deliberately changed during group motion between successive checkpoints.

5.1.2 Specification of Parameters Used in the Implementation

The nodes in the network are also pre-configured with knowledge of their GIDs; thus, group members are supplied with knowledge of their assemblage at start-up. Although the implementation of the protocol can infer the dissociation and merging of groups, the dynamism of group memberships has not been simulated, i.e., the affiliations of nodes remains static throughout the simulations. Table 5.1 lists the values assigned to some parameters of the routing protocol in the implementation.

The FULL_DUMP_UPDATE_PERIOD and INCREMENTAL_UPDATE_PERIOD denote the time periods between successive *full dumps* and *incremental updates* broadcasted by the intra-

Table 5.1: Values of parameters used in implementation

Parameter	Value
FULL_DUMP_UPDATE_PERIOD	14.0 secs
INCREMENTAL_UPDATE_PERIOD	1.0 secs
NEIGH_TIMEOUT_PERIOD	3.5 secs
GRP_DISSOCIATION_PERIOD	20.0 secs
ROUTE_EXPIRATION_PERIOD	5.0 secs
REPLY_TIMEOUT_PERIOD	7.0 secs
GID_MAP_CACHE_PERIOD	15.0 secs

group routing protocol at each node respectively. Their values are the same as that of the ns-2 implementation of DSDV. NEIGH_TIMEOUT_PERIOD is the amount of time a node must wait to hear of a successive routing broadcast from its neighboring node, failing which, it deems the link to the neighbor to be down. When a node does not hear of a routing update from another group member for GRP_DISSOCIATION_PERIOD seconds, it assumes that the latter has dissociated from the group. This value is nominally set at 20 seconds: the simulation of dynamic memberships has not been performed. ROUTE_EXPIRATION_PERIOD refers to the time duration for which any node will maintain information about a route to another group in the network in its cache. Likewise, the GID_MAP_CACHE_PERIOD refers to the time period through which any node will maintain the mapping between node address and GID of nodes belonging to other groups in its cache. GID mappings and routes to other groups in the network are cached from inter-group control packets that are being forwarded by nodes. These values have been set as 5 seconds (by experimentation) and 15 seconds (nominally). The REPLY_TIMEOUT_PERIOD refers to the amount of time that the source of a RREQ will wait for a RREP from the destination before re-initiating a RREQ in the network.

A total of 285 experiments were conducted on a range of mobility and connection patterns. The general goal of the experimentation was to, firstly, demonstrate that group mobility presented a more conducive operating environment for all routing protocols and GMRP was tailored to this environment; secondly, to determine the network scenarios that corresponded to the crest-and-trough in the performance of GMRP; and finally, to verify if the performance of GMRP is within reasonable bounds of conventional MANET routing protocols like AODV and DSDV in the absence of groups in the network. The experiments have been organized under 6 different categories. The parameters used in the modeling of the network have been tabulated separately for each experimental category, and the associated results are presented in the form of graphs. It is noted that some of the results reported for DSDV have been extrapolated from those of shorter runs when the simulations did not run for the entire time duration that was desired. The traffic load in Experiments 1–5 consists of TCP connections. Each TCP connection is fed by an ftp agent at the application layer which continuously sends new data to the TCP agent. Every TCP connection lasts throughout the simulation duration.

5.2 Experiment 1: Studying the Effect of the Connection Mix

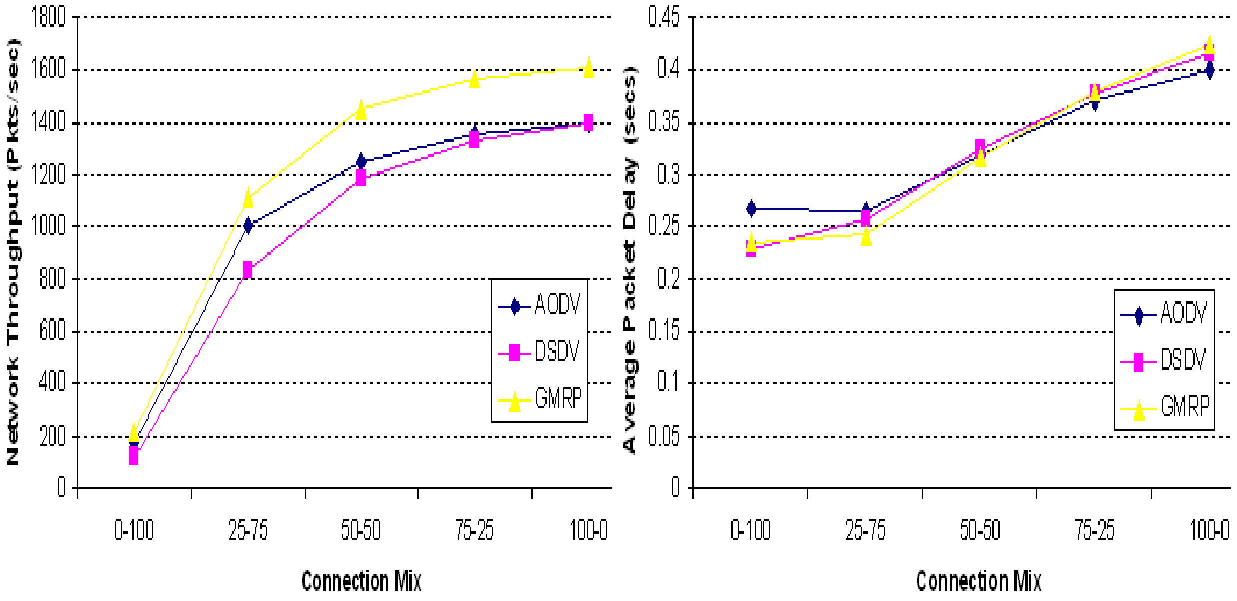
This experiment seeks to establish the effect of varying the proportions of intra- and inter-group connections in the overall load on the network. The experiments were conducted on 5 different network scenarios: A, B, C, D and E; the parameters that describe the experimental setup of each scenario are enumerated in Table 5.2. The transmission range of the wireless nodes is set to the default value of 250m. The simulation duration of each experiment was 1000 seconds. For each network scenario, the network throughput and average packet delay of AODV, DSDV and GMRP were compared while varying the ratio of intra- and inter-group connections in the overall traffic load. The variation of network throughput and average packet delay with different connection mixes are shown in Figures 5.1(a) and 5.1(b) respectively, based on the average results from the 5 network scenarios.

In Figure 5.1(a)and 5.1(b), a label *a–b* in the X-axis denotes the proportion of intra- and inter-group traffic and is to be interpreted as follows: "*a*% of the total traffic load is made of intra-group connections while the remaining *b*% is of inter-group connections". As shown in Figure 5.1(a), the throughput gain from employing GMRP at the routing layer increases with more intra-group bias in the traffic load.

Table 5.2: Parameters used in creation of simulation scenarios for Experiment 1

Parameter	Value				
Scenario Label	A	B	C	D	E
Wireless range (*m*)	250	250	250	250	250
Number of nodes	200	300	400	500	600
Number of groups	10	15	20	25	30
Nodes per group	20	20	20	20	20
Field area (*m*×*m*)	4000×4000	4500×4500	5000×5000	5500×5500	6000×6000
Simulation time (*secs*)	1000	1000	1000	1000	1000
Type of connections	TCP	TCP	TCP	TCP	TCP
Number of connections	100	150	200	250	300
Node velocity (*m/s*)	15	15	15	15	15

This gain attains a saturation when the traffic is comprised of about 75% of intra-group connections. When the traffic load is purely inter-group based, the throughputs of AODV and GMRP are quite similar, and marginally better than DSDV. The savings from the use of controlled flooding to search for the destination in GMRP are offset by the periodic broadcasting of intra-group updates which do not contribute to any throughput in these circumstances. The throughputs of DSDV evince a late upsurge when the communication load is more intra-group based. This points to DSDV's ability to maintain routes to nearby nodes with greater ease than to nodes that are farther away: updates about farther nodes take longer times to propagate when routing broadcasts are made periodically by unsynchronized nodes. The use of the ring-search mechanism in AODV seems to benefit a purely intra-group traffic load. According to the ring-search mechanism, AODV first searches for the destination within the first *n* hops from the source, failing which, the search is expanded to a larger ring around the source, and so on, until the entire network is to be searched. This *expanded ring-search* works well with intra-group connections because AODV may be able to fulfill a route acquisition by searching only a portion of the network, and not deluging the entire set of nodes. This may be the reason why AODV is able to perform as well as DSDV when the load is purely intra-group oriented. Overall, the diminished throughput of DSDV can be attributed to the control overhead from having to maintain routes to the rest of the network at all nodes at all times during the simulation.



(a) Network Througput (b) Average Packet Delay

Figure 5.1: Effect of Connection Mix on Protocol Performance (Average)

The curves illustrating the average delay suffered by a packet when deploying either protocol are quite similar beyond the 25–75 mark. As mentioned before, the average packet delay is calculated only for those packets that have been successfully delivered. Thus, packets that were not delivered by the end of the simulation or dropped were not taken into account. Thus, the significance of the average packet delay as a standalone metric must not be overstated. For instance, the increased latency suffered by packets when using GMRP than AODV in a 100% intra-group load may be from the additional packets delivered by GMRP over AODV.

The most important factors that could prolong the delivery latency of a packet using GMRP are:

1. GMRP uses a reactive inter-group routing mechanism, and therefore incurs a route acquisition delay during inter-group communications.

2. The inter-group routes in GMRP may be sub-optimal. Longer paths lead to larger propagation delays.

In light of the above factors it was anticipated that GMRP would lead to longer packet delays than DSDV in inter-group communication scenarios and the delay graphs suggest that too; yet, this difference is only marginal. When the load is purely inter-group based, the throughputs of AODV, DSDV and GMRP are fairly similar. The delay graphs show that AODV and GMRP suffer slightly longer delays than DSDV owing to larger route acquisition delays. The use of the *expanded ring-search mechanism* in AODV may lead to extraordinarily long delays in route acquisitions to destinations lying in the very edges of the network. This is possibly the reason why AODV shows the worst average packet delay between the 3 protocols in the 0–100 mark. As the percentage of intra-group connections in the traffic load increases, the average packet delays of all protocols tend to converge.

5.3 Experiment 2: Studying the Effect of Traffic Loads

In this category of experiments, the 3 protocols, operating in 5 different network scenarios of sizes 200, 300, 400, 500 and 600 nodes, were subjected to varying traffic loads. The other parameters defining the simulation scenario have been listed in Table 5.3. In Figures 5.2 and 5.3, the traffic load on the network (the values on the X-axis) represent the number of connection pairs as a percentage of the total number of mobile nodes in the network. Thus, the experiment using scenario A in Table 5.3 was repeated with traffic loads of 30, 60, 90, 120 and 150 TCP connections. Figure 5.2 shows the network throughput and average packet delay results averaged over the 5 network scenarios A, B, C, D and E for increasing traffic loads which are evenly distributed between intra-group and inter-group connections. That is, the ratio of intra- and inter-group connections is 50–50. Also, a second set of experiments were performed in which the communicating pair of nodes was chosen in a random manner. The performance results for the 3 protocols when connection pairs are chosen randomly are presented in Figure 5.3. This set of experiments were performed on only one network scenario, namely A.

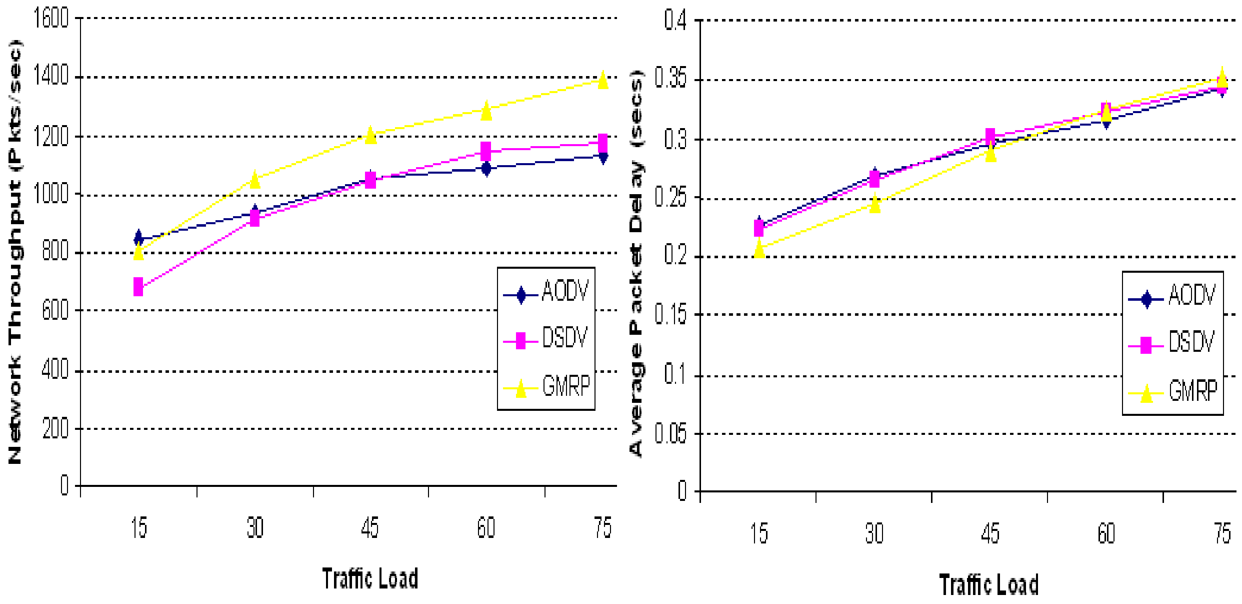
Under light traffic loads (15% mark), the network throughput realized from AODV is observed to surpass that of GMRP and DSDV in both sets of experiments. AODV performs considerably better than the other protocols around this region since a fewer number of route acquisitions are required in the face of small traffic loads. As the load on the network increases through 45% on to 75%, the throughput of AODV plumets to well below that of GMRP and DSDV in Figure 5.2(a). Likewise, in the case of random connections, the throughput of DSDV is observed to be better than that of AODV at the 60% mark. AODV is unable to cope with heavy traffic loads because the network is inundated with repeated RREQ messages that must be broadcasted to acquire routes. In the first set of experiments (Figure 5.2(a)), the performance difference between DSDV and GMRP is constant until the 60% mark, after which GMRP exhibits an increased performance

Table 5.3: Parameters used in creation of simulation scenarios for Experiment 2

Parameter	Value				
Scenario Label	A	B	C	D	E
Wireless range (*m*)	250	250	250	250	250
Number of nodes	200	300	400	500	600
Number of groups	10	15	20	25	30
Nodes per group	20	20	20	20	20
Field area ($m \times m$)	4000×4000	4500×4500	5000×5000	5500×5500	6000×6000
Simulation time (*secs*)	1000	1000	1000	1000	1000
Type of connections	TCP	TCP	TCP	TCP	TCP
Proportion of intra-inter connections	50–50	50–50	50–50	50–50	50–50
Node velocity (*m/s*)	15	15	15	15	15

gain. The control overhead generated by DSDV is independent of the traffic load, and hence its network throughput shows a steady increase with traffic load when compared with AODV. The volume of intra-group routing overhead of GMRP is independent of the traffic load on the system; the route requests sent by the inter-group routing module, however, depend on the number of such connections initiated by the network. The immunity of GMRP to increased traffic loads is indicative of the extent of savings in overhead from the contained flooding of RREQ messages in the inter-group routing module. Again, GMRP is able to perform considerably better than DSDV because the volume of control overhead generated by GMRP is significantly less than of DSDV. The network throughput gain of GMRP over DSDV and AODV rises with increased number of network connections and peaks at the 75% mark.

The average delay suffered by a packet increases with the number of network connections in Figure 5.2(b) for all protocols. This could be from increased channel contention, introduction of connections that fall along longer paths etc. The delay-wise performance of GMRP is the best amongst the protocols until the 45% mark, beyond which AODV shows a marginal betterment in



(a) Network Througput

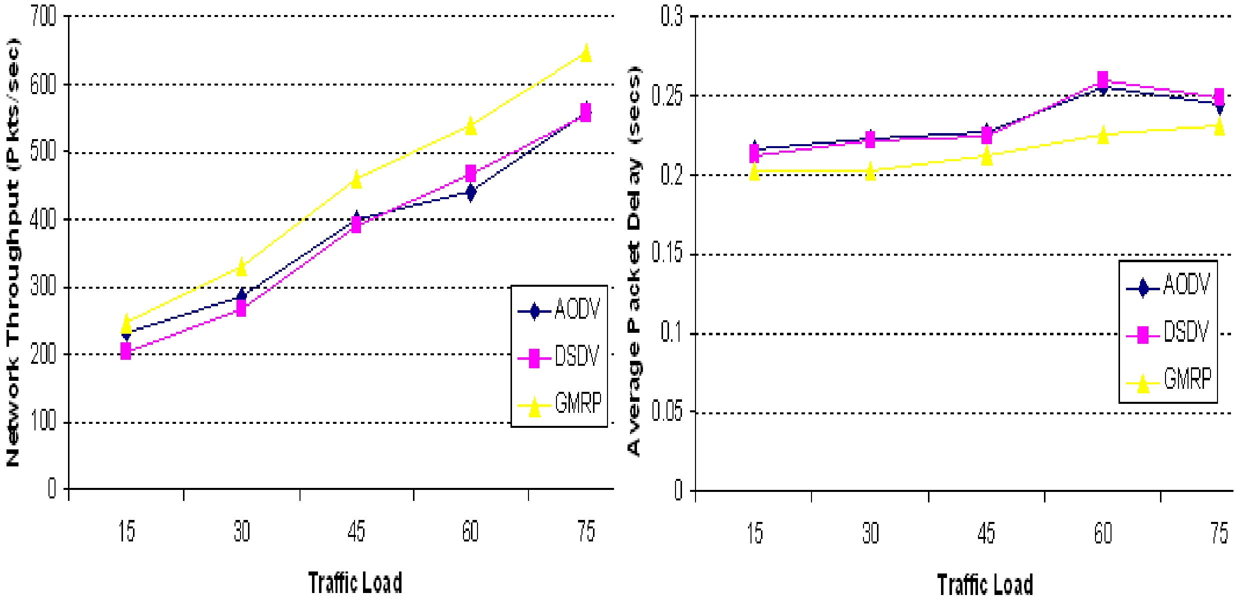
(b) Average Packet Delay

Figure 5.2: Effect of Traffic Load on Protocol Performance (Mixed Connections—Average)

performance. The delay curves in Figure 5.3(b) compare the average delay suffered by a packet when the communicating pairs of nodes are chosen randomly. GMRP demonstrates a mild gain in delay-wise performance in comparison with the other 2 protocols. This gain is also observed to increase with an increase in the number of connections: this conforms to the increased throughput gains over AODV and DSDV in this region.

5.4 Experiment 3: Studying the Effect of Group Sizes

The performances of the 3 protocols in MANETs of different group sizes are studied in this category of experiments. 5 network scenarios of sizes 120, 240, 360, 480, 600 were created using the parameters listed in Table 5.4. For each network scenario, the performance of each protocol was assessed while changing the number of groups in the network through 2, 4, 6, 8, 10 and 12 under the same traffic load. Thus, the network of size 480 was simulated on groups of size 40, 48, 60,



(a) Network Througput

(b) Average Packet Delay

Figure 5.3: Effect of Traffic Load on Protocol Performance (Random Connections—200 Nodes)

80, 120 and 240 nodes, for an example. Further, 3 sets of experiments were performed on each network scenario depending on the connection mix of the traffic: 100% intra-group connections (Set 1), 100% inter-group connections (Set 2), and a 50–50 distribution between intra- and inter-group communications (Set 3). This leads to a total of 18 experiments against each network (6 different group sizes for 3 types of traffic) and 90 experiments overall (5 networks), whose results are reported through Figures 5.4–5.6.

The objective of this category of experiments is to find the equilibrium point between the intra- and inter-group routing overhead of GMRP over the group size. A larger group size leads to more intra-group routing entries and thereby more intra-group routing overhead; yet, the number of paths that an inter-group route request must take to cover the rest of the network may then be smaller. Conversely, smaller groups lead to relatively less intra-group routing overhead but more branches in the network where route requests must be propagated to fulfill inter-group route

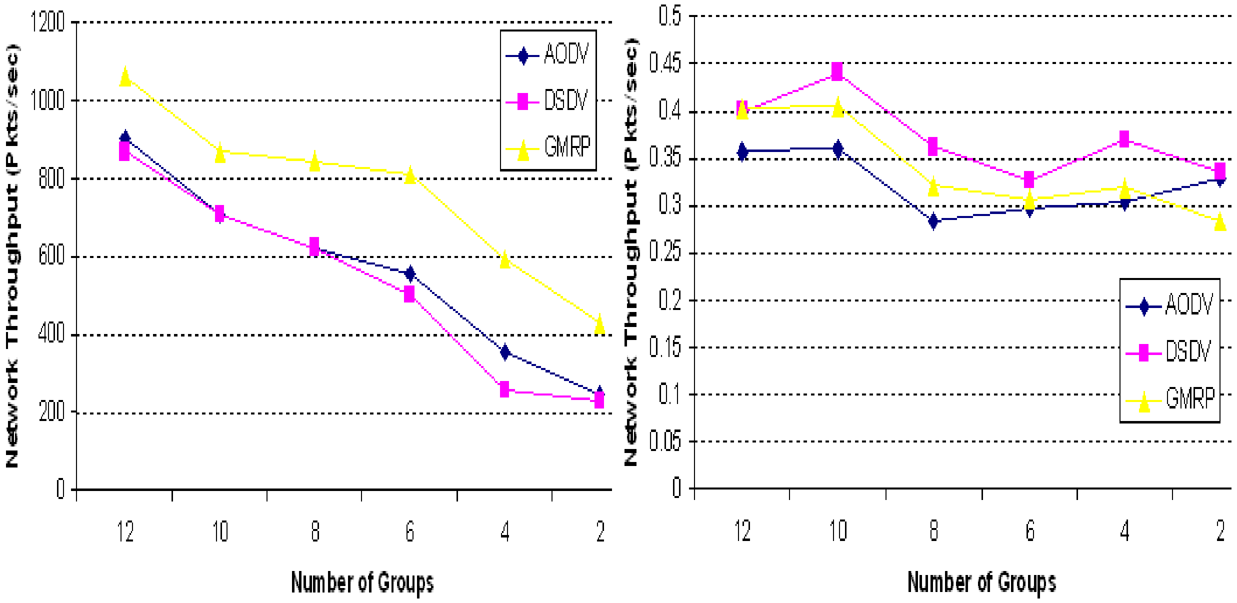
Table 5.4: Parameters used in creation of simulation scenarios for Experiment 3

Parameter	Value				
Scenario Label	A	B	C	D	E
Wireless range (*m*)	250	250	250	250	250
Number of nodes	120	240	360	480	600
Field area ($m \times m$)	3500×3500	4000×4000	4500×4500	5000×5000	5500×5500
Simulation time (*secs*)	600	600	600	600	600
Type of connections	TCP	TCP	TCP	TCP	TCP
Number of connections	60	120	180	240	300
Node velocity (*m/s*)	15	15	15	15	15

acquisitions.

The network throughputs achieved by AODV, DSDV and GMRP on a purely intra-group traffic load are depicted in Figure 5.4(a). As the group size increases, the network throughput obtainable from either protocol falls down. An increased group size means that the traffic load on a group is increased too. The fact that more connections are concentrated within a group leads to more intersections in the connection paths (whose average length increases with increasing group sizes) thereby aggravating channel contention in the network. This is the primary reason for the plummet in the *network throughput* of all 3 protocols with increased group size.

On the contrary, the throughput curves of all 3 protocols increase with group size in Figure 5.5 where the traffic is purely inter-group based. This increase is sustained until the number of groups reaches 4, after which the network throughput falls. This is because of an increased availability of inter-group connections with fewer groups in the network. Recapitulating on the mobility model for the motion of distinct groups in the network, each group is assigned an initial position in the network, and goes on to choose a new checkpoint to which it must move. With only 6 groups in the network, there are only 6 checkpoints, between which either group shuttles. This results in inter-group connections being more available than when the network has 6 groups. Further, with



(a) Network Througput

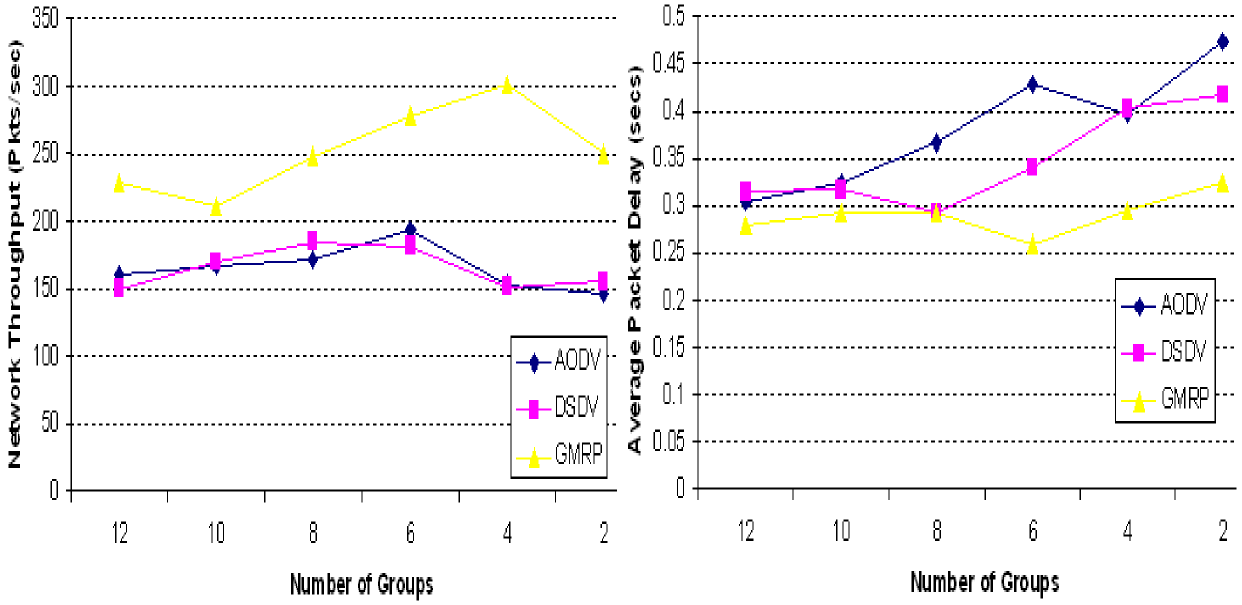
(b) Average Packet Delay

Figure 5.4: Effect of Group Size on Protocol Performance (Intra-group Connections)

larger and hence fewer groups in the network, the number of relative inter-group movements in the network is reduced leading to more connection stability. However, when the number of groups is reduced further, throughput falls from denser spatial concentration of connections despite the greater availbility of connections.

In Figure 5.6(a), the throughput curves for mixed connections fall with larger group sizes. This fall is because the overall throughput is more influenced by the intra-group throughput values than inter-group throughput (Intra-group throughput values in Figure 5.4 are significantly better than inter-group throughputs of Figure 5.5(a)); yet, the fall is not as steep as of Figure 5.4(a).

Next, a comparison in the performances of the 3 protocols, and the specific protocol characteristics affected by group size are analyzed. GMRP outperforms AODV and DSDV in all cases in Figure 5.4(a). The fact that DSDV maintains routes unnecessarily to nodes in other groups in the network results in its poorer performance. The performance gain from GMRP over AODV



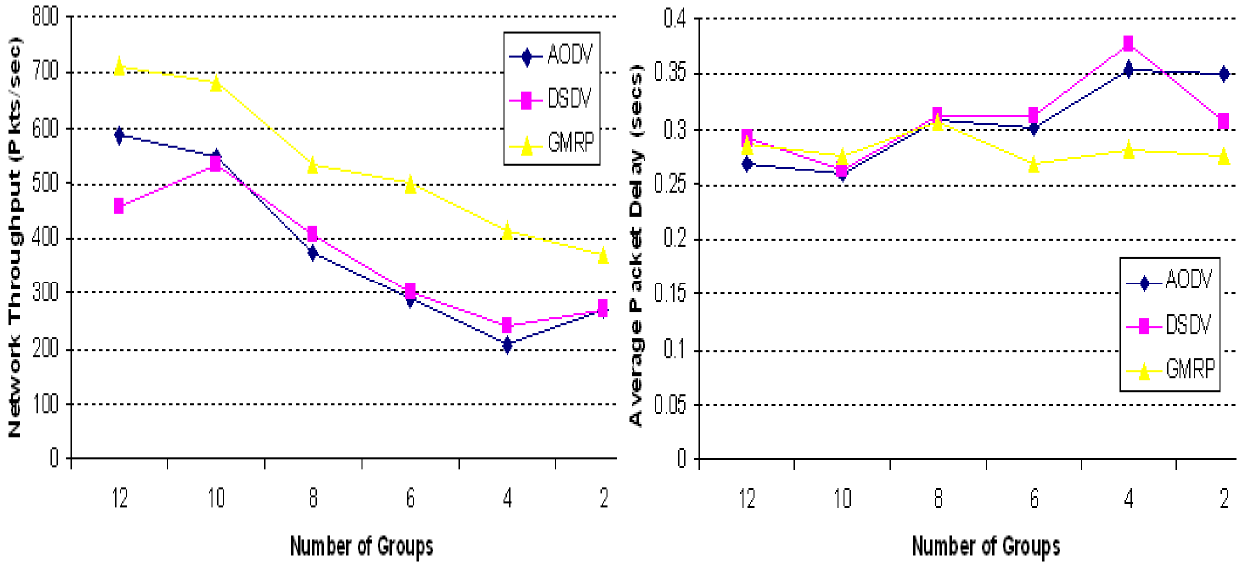
(a) Network Througput

(b) Average Packet Delay

Figure 5.5: Effect of Group Size on Protocol Performance (Inter-group Connections)

increases as the number of groups decreases from 12 to 6. As the size of the group increases, the concentration of more connections within a group leads to a more localized deluge in route request messages from the ring-search mechanism of AODV. With larger groups that have higher traffic loads, more rings in the search mechanism of AODV are vulnerable to intersect, thereby causing more channel contention. The throughput of DSDV closely matches that of AODV in this phase. As the number of groups is further decreased to 2, the performance gain of GMRP over AODV falls back a little bit. Large group sizes are less conducive to GMRP because the volume of intra-group routing updates is large. It is noted that the volume of control overhead generated by DSDV is irrespective of the size of a group: DSDV maintains routes to all network nodes.

In Figure 5.5(a), the throughput gain of GMRP over AODV increases as the number of groups in the network is reduced from 12 to 4. Smaller groups incur more network paths through which GMRP must search for an inter-group destination. As the group size increases, the number



(a) Network Througput (b) Average Packet Delay

Figure 5.6: Effect of Group Size on Protocol Performance (Mixed Connections)

of paths that GMRP searches for the destination is reduced, thereby reducing the associated overhead. However, the savings from more contained RREQ propagation paths are offset by increase in the volume of intra-group routing updates, which leads to the reduction in performance gain at the 2-mark. The maximum performance gain is hence yielded between the 6 and 4 points. The throughput curves in Figure 5.6 reiterate that throughput gain of GMRP over AODV and DSDV is maximal near the middle regions of the X-axis (that is between the 8 and 4 marks). In summary, small and large group sizes incur relatively more inter-group and intra-group routing overhead for GMRP respectively and the peak performance gain of GMRP is observed in the central regions of the X-Axis.

The delay graph in Figure 5.4(b) shows that the average packet delay drops with the increase in group size in the case of a purely intra-group traffic load. However, a larger group radius will elongate the average length of an intra-group route thereby increasing the propagation time of a packet: the experimental results seem to contradict this expectation. The observed fall in delay with increasing group sizes is also accompanied by a plummeting of the network throughput. A possible explanation for the fall in packet delay with larger groups is the fact that many of the packets that were initially transmitted were not delivered and hence did not contribute to the computation of the average packet delay. The delay graphs show that AODV has the least value for the average packet delay measured for intra-group connections. The delay curves in Figures 5.5 and 5.6 show that GMRP performs better than AODV and DSDV, and that the savings in the *average packet delay* are a maximum when the number of groups in the network is between 6 and 4.

5.5 Experiment 4: Studying the Effect of Node Velocities and the Rate of Group Topological Changes

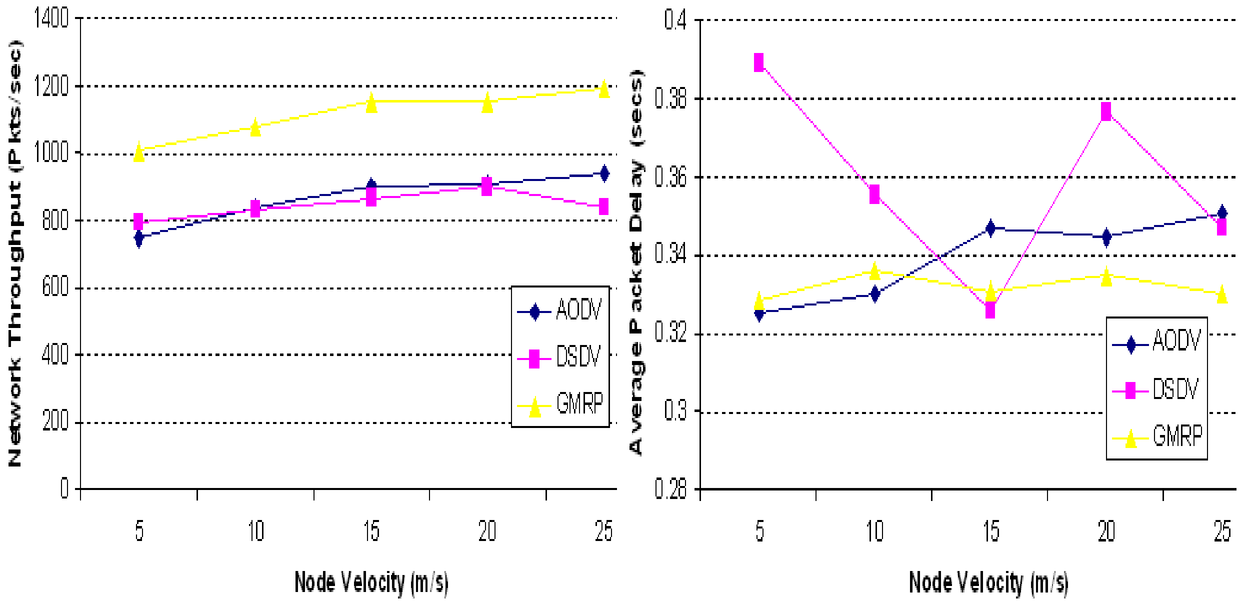
This category of experiments studies the effect of varying the parameters that define the mobility patterns of groups and group members in a MANET. Firstly, the mobility model of group members is altered to induce more topological changes in a group. Leading up to this experiment, the mobility model imparted a change in the spatial configuration of groups every time a new checkpoint was reached. Once a group reaches a checkpoint, a new checkpoint is sought to which all group members will move towards. Within the *group area* defined by this target checkpoint, group members randomly choose new grid positions that each will move into leading to a change in spatial configuration. The amount of topological dynamism defined by these rules is moderate when checkpoints are separated by a considerable distance. In the current set of experiments, additional topological dynamism is incorporated by requiring group members to change their spatial configurations once in *t* seconds. Specifically, at the end of every *t*-second epoch since the last checkpoint attained by the group, every group member re-computes a new grid position that it must move into, within the group area formed about the projected trajectory of the group at the expiration of the next *t* seconds. This leads to a more fragile intra-group mobility model than used through the previous experiments and is denoted as the *weak mobility model* as against the *strong mobility model* used before. The value of t is set as 40 seconds for the weak mobility model.

Table 5.5: Parameters used in creation of simulation scenarios for Experiment 4

Parameter	Value				
Scenario Label	A	B	C	D	E
Wireless range (m)	250	250	250	250	250
Number of nodes	200	300	400	500	600
Number of groups	10	15	20	25	30
Nodes per group	20	20	20	20	20
Field area ($m \times m$)	4000×4000	4500×4500	5000×5000	5500×5500	6000×6000
Simulation time (*secs*)	1500	1500	1500	1500	1500
Type of connections	TCP	TCP	TCP	TCP	TCP
Number of connections	120	180	240	300	360
Proportion of intra-inter connections	50–50	50–50	50–50	50–50	50–50

The second mobility parameter that is varied is the node velocity. Node velocity influences the rate at which the statuses of links between neighboring groups go up-and-down. It also influences the *link change rate* within a group when group members are programmed to change configurations.

5 networks of sizes 200, 300, 400, 500 and 600 nodes were created based on the parameters listed in Table 5.5. Group members in each network were subject to motion under the constraints of the strong and weak mobility models. Further, node speed of each network is varied from 5 m/s to 10, 15, 20 and 25 m/s in both the strong and weak mobility models. Thus, a total of 10 experiments were conducted on each network (5 node speeds corresponding to 2 mobility models), and 50 experiments, in all, in this category (5 network scenarios A, B, C, D, E in Table 5.5). The throughput and delay curves for the 5 node speeds of the weak mobility model are presented in Figure 5.7, while those of the strong mobility model are shown in Figure 5.8. All data points are



(a) Network Througput

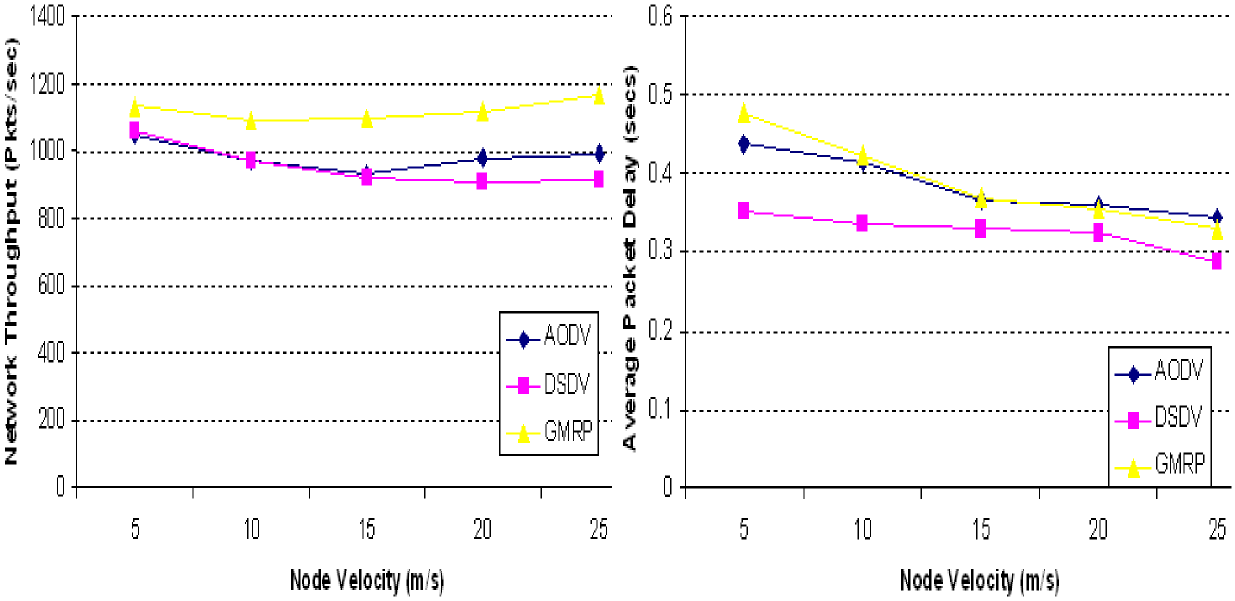
(b) Average Packet Delay

Figure 5.7: Effect of Node Speed on Protocol Performance (*Weak Mobility Model*)

averaged over the 5 different networks.

From the throughput curves in Figures 5.7(a) and 5.8(a), it is observed that the strong mobility model yields a higher throughput than the weak mobility model for all node speeds. More frequent topological changes within the group leads to frequent link and connection failures in the weak mobility model. The control overhead required in repairing failed connections results in reduced network throughput.

The graphs also show that the network throughput increases with increase in node speeds for both mobility models for GMRP. The throughput curves of DSDV and AODV climb up with increasing speeds for the weak model, but fall back a little in the case of the strong mobility model. It is noted that these observations were not consistently recorded with all 5 network scenarios: a fall in network throughput of all 3 protocols was observed on some network scenarios. The increase in network throughput with node speeds is counter-intuitive: increased nodal mobility leads to a



(a) Network Througput

(b) Average Packet Delay

Figure 5.8: Effect of Node Speed on Protocol Performance (*Strong Mobility Model*)

higher *link change rate* in the network and hence incurs more routing overhead. [17] and [43] report a drop in the network throughput, while [18] observes a decrease in the *average path duration*[3] of the network with increase in node velocities. The contradictory trend in the results presented in Figures 5.7(a) and 5.8(a) is not influenced by the presence of groups: increased node mobility does result in lesser intra-group link sustenance too. One probable explanation to the anomaly is the increased availability of inter-group connections resulting from groups coming within the communication range of one another on a more frequent basis. Depending on the exact trajectories of network nodes, network partitioning and the isolation of some nodes from slow nodal movement is conceivable. Thus, it is possible that even though increased node speeds may result in frequent changes in link statuses and therefore increase the routing overhead of a protocol, it may also bring more network nodes within the communication plane of one another. It is also probable that if

[3] The average amount of time for which the communication path between any source-destination pair is expected to last. The *shortest path* between a source and destination is regarded as the communication path between the pair at a particular instant of the network lifetime.

the velocities were increased any further, the network throughput will have fallen down. Overall, the effect of node velocity on the network throughputs of either protocol is inconclusive; both increased and decreased network throughputs with increasing node speeds were observed under different network scenarios.

The throughput-wise performance of all 3 protocols increases at a faster rate with increase in the node speed in the weak mobility model than the strong mobility model. The reduction in the performance gains of the protocols in the strong mobility model over the weak mobility model from the 15m/s to 25 m/s mark suggests that the throughput attained in this region is close to the saturation point of network: even if the topology of groups were to be completely static, any further performance gain would be contingent upon factors such as channel contention.

The difference in the throughputs of AODV and GMRP is more-or-less constant with varying node speeds for both mobility models. The performance of both GMRP and AODV on a purely inter-group connection scenario are fairly similar (refer to Figures 5.1(a)). Assuming that the increased throughputs of AODV and GMRP with increasing node speeds is due to the availability of more inter-group connections, the similarity in the throughputs of AODV and GMRP when faced with purely inter-group connections may explain the constant difference in the throughputs of AODV and GMRP with varying node speeds. The throughput of DSDV wanes below the level of AODV at high node speeds. Overall, there isn't conclusive evidence to suggest that the performance gain of GMRP over AODV may be dependent on node speed or the choice of the mobility model, and in turn the average rate of change in link statuses.

The average packet delay is expected to increase with increasing node speeds, based on the assumption that more inter-group connections along longer paths (than intra-group connections) are being made available with increased node speeds. This trend holds true for the weak mobility model. Yet, in the case of the strong mobility model, the average packet delay is observed to fall a notch with each increment of node speed. The differences in the average packet delay suffered by AODV and GMRP are practically small. DSDV shows the best delay-wise performance for the strong mobility model. Overall, the delay-wise performance comparison between the 3 protocols has no correlation with the mobility pattern or speeds of the nodes.

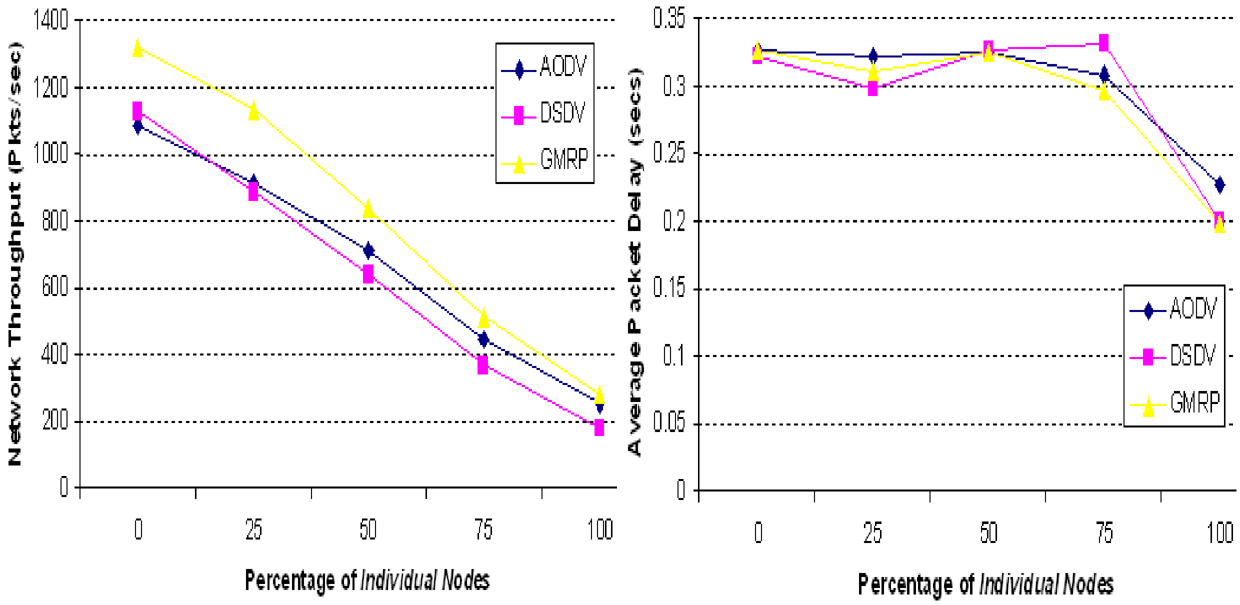
5.6 Experiment 5: Studying the Effect of the Presence of *Individual Nodes*

This set of experiments compares the performances of MANETs that contain different percentages of *individual nodes* when using either of the 3 candidate routing protocols. It aims to assess the suitability of GMRP to MANETs that contain a substantial percentage of *individual nodes*, especially since all groups are formed from nodes that initially move about independently.

Table 5.6: Parameters used in creation of simulation scenarios for Experiment 5

Parameter	Value				
Scenario Label	A	B	C	D	E
Wireless range (*m*)	250	250	250	250	250
Number of nodes	200	300	400	500	600
Nodes per group	25	25	25	25	25
Field area ($m \times m$)	4000×4000	4500×4500	5000×5000	5500×5500	6000×6000
Simulation time (*secs*)	1500	1500	1500	1500	1500
Type of connections	TCP	TCP	TCP	TCP	TCP
Number of connections	100	150	200	250	300
Node velocity (*m/s*)	15	15	15	15	15

5 network scenarios of sizes 200, 300, 400, 500 and 600 nodes were created based on the parameters listed in Table 5.6. The percenatge of individual nodes on each network scenario was varied through 0%, 25%, 50% 75% to 100%. Thus, the number of individual nodes in network scenario C (size 400) was increased from 0 to 100, 200, 300 and 400 nodes. The remaining nodes in the network were organized into groups of size 25. Thus, the number of groups in network

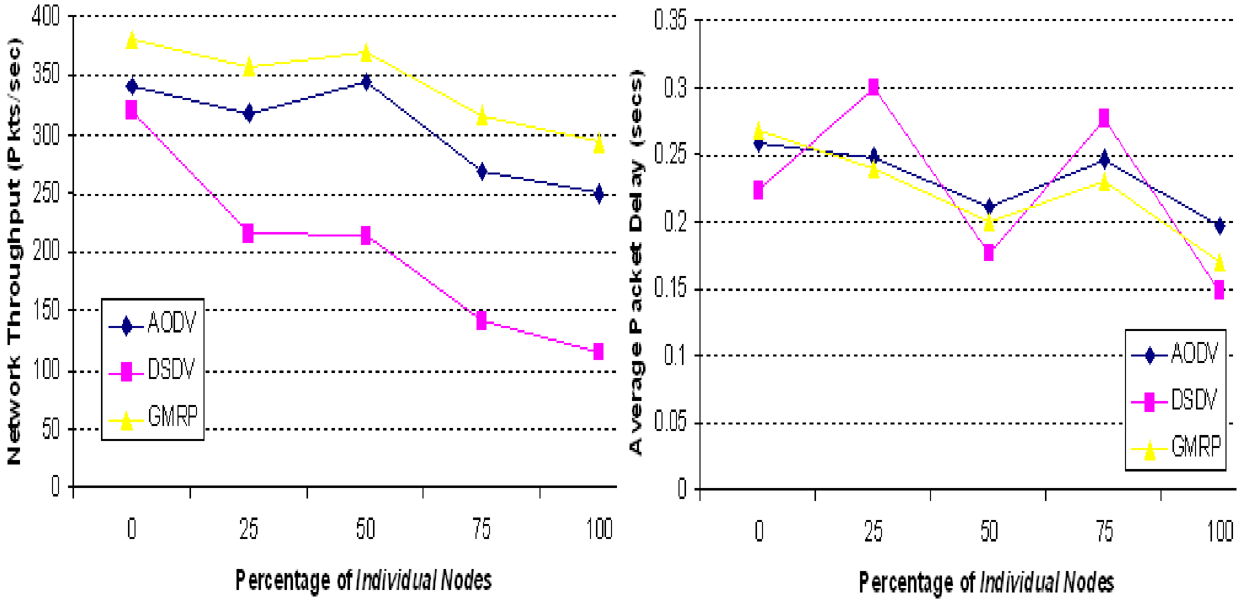


(a) Network Througput (b) Average Packet Delay

Figure 5.9: Effect of Presence of *Individual Nodes* on Network Performance (Mixed Connections—Average)

scenario C was decreased from 16 to 12, 8, 4 and 0. The performance of each network is tested on 2 kinds of connection scenario: Set 1 and Set 2. The traffic connections in Set 1 were of 2 kinds: a percentage of them, equal to the percentage of individual nodes in the network originated and/or terminated at an individual node, while the remaining were distributed between group members. Of the group-based connections, 50% of connections were intra-group connections while the other 50% were inter-group connections. The second set of experiments were conducted only on the network of size 300 nodes. In this set, 150 TCP connections were setup between randomly chosen communicating pairs. The experimental results for Set 1 and Set 2 are reported in Figures 5.9 and 5.10 respectively.

When the network is entirely made up of individual nodes, GMRP is made to behave as a purely reactive protocol that regards each node as a group. Thus, the periodic intra-group routing broadcasts made by each node only contain routing information to its neighboring nodes (groups).



(a) Network Througput (b) Average Packet Delay

Figure 5.10: Effect of Presence of *Individual Nodes* on Network Performance (Random Connections—200 Nodes)

It is noted that GMRP may also operate as a purely proactive protocol by considering the entire network as a single group.

The throughput graphs in Figures 5.9(a) and 5.10(a) suggest that the network throughput decreases with the introduction of more individual nodes into the network. This corroborates the observation in [17] and [42] that group-oriented motion of nodes provides for a more topologically stable network whose performance may be considerably better than of a network comprising independently moving nodes.

The network throughputs derived from the random choice of connection pairs in Figure 5.10 is much less than from the connection scenarios that necessarily involve group members in a network of size 300 nodes: the necessary inclusion of intra-group connections in Set 1 holds for a substantial portion of the throughput obtained in this set of experiments. For the same reason, the fall in throughput with the addition of individual nodes (and thereby the percentage of connections involving them) is much less in the experiments in Set 2 than Set 1[4].

The performance gain from employing GMRP against AODV decreases with the introduction of more individual nodes in the network. This is reflected in the throughput curves of both sets of experiments. In Set 1 (Figure 5.9(a)), the performance curves of GMRP and AODV taper inwards to converge when the network contains no groups. The addition of individual nodes (and the accompanying reduction of intra-group connections) reduces the margin of gain from applying GMRP, as expected. When the network contains only groups, DSDV performs a trifle better than AODV on the average as reported in Figure 5.9(a). The throughput performance of DSDV falls beneath that of AODV when the percentage of individual nodes is increased because the unrelated motion of individual nodes proliferates the number of triggered routing updates that DSDV makes.

The average packet delay suffered by either protocol decreases with an increased presence of individual nodes for both sets of experiments. This can mainly be attributed to the reduction in the number of delivered packets that are factored into the calculation of the value, and differences in the average length of connection routes between experiments. The differences in delay between AODV and GMRP are minimal; the delay curves for DSDV are zigzagged.

In summary, this category of experiments confirms that all 3 routing protocols perform better when faced with networks that contain groups than otherwise. Further, the performance of GMRP scales well with rising proportions of individual nodes and is quite as good as AODV in a MANET that has no groups.

From the 225 experiments conducted on various kinds of mobility and traffic scenarios as described through Sections 5.2–5.6, the average network throughput obtained from employing GMRP is calculated to be 25% over the mark of AODV and 30% better than DSDV. Further, the mean of the average packet delay from 225 experiments for GMRP is 1% better than of DSDV,

[4] The fall in network throughput for 300 nodes in Set 1 is similar to the throughput curves in Figure 5.9(a).

and about 3% better than of AODV.

5.7 Experiment 6: Comparing the Route Acquisition Characteristics of the 3 Protocols

In this section, experiments are carried out to compare the route acquisition characteristics, namely, the *route acquisition delay* and the *number of fulfilled connections*, of the 3 protocols. The route acquisition delay of a protocol for a connection is defined as the latency between the time at which the connection is sought and the time of reception of the first packet at the destination. Thus, route acquisition delay is calculated as the sum of the time taken to acquire a route to the destination and the propagation delay of the first packet of the connection. Towards the comparison of the route acquisition delay between the 3 protocols, an average value is computed over all the connections set up in the simulations. The number of fulfilled connections is computed as the number of connections for which the first packet is successfully delivered at the destination.

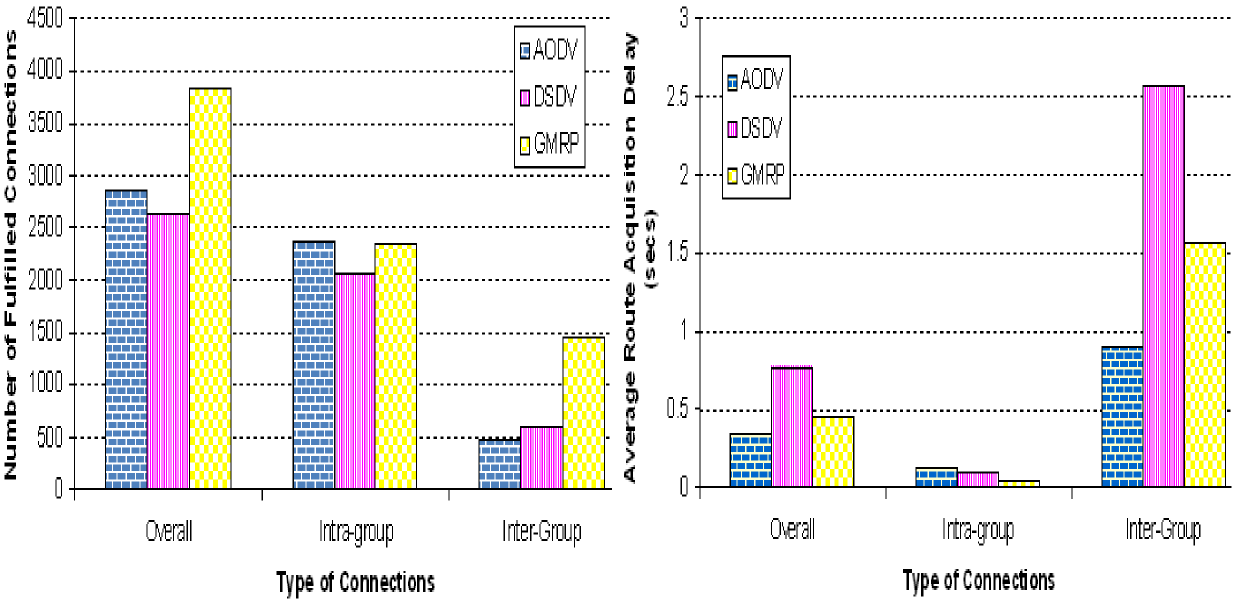
In the previous experiments, the average packet delay was calculated as the average of the packet delays suffered by all packets that were successfully delivered during the simulation. While the amount of time required by the protocol to acquire a route to the destination does factor into the computation of the average packet delay, its magnitude of significance is trivial. Since all the connections set up in the previous experiments involved TCP flows fed by ftp agents, a connection establishment was followed up by the delivery of a large number of packets. Connections usually lived through the duration of the simulation. Thus, the route acquisition delay factored into the average packet delay only so much as the frequency of route faliures (from nodal movement) in the network. Thus, the average packet delay was more significantly affected by the average length of routes used by the protocol and the channel contention in the network than the route acquisition delay. This is probably why the there was little to choose between the 3 protocols in the previous experiments, although, as has been repeatedly stated, GMRP does employ longer routes than the other protocols, and, the 3 protocols consume different proportions of network bandwidth for route acquisitions under different circumstances (which affects channel contention). A comparison of the route acquisition delays between the 3 protocols is expected to yield more information on the sheer swiftness at which a connection can be established. Yet, because the route acquisition delay also includes the propagation delay of the first packet, route length and channel contention have a small weightage on the computation.

30 network scenarios representing different network sizes, node speeds, mobility models and group sizes were created with parameters similar to those described in the previous experiments. The wireless transmission range was retained at 250m. Each network (mobility) scenario was subjected to 2 kinds of traffic loads: 100 intra-group connections and 100 inter-group connections. Each connection required the transmission of only 1 data packet of size 1000 bytes following which the connection was terminated. Each protocol was thus run on 60 simulation scenarios (30 mobility scenarios subjected to 2 kinds of connection scenarios each). The simulation duration of each experiment was 100 seconds and all connections were started within the first 50 seconds of the simulation. The comparison of the number of fulfilled connections is reported in Figure 5.11(a), while the route acquisition latency of the 3 protocols is compared in Figure 5.11(b).

Since AODV is a purely on-demand routing protocol, any connection request will suffer a latency before AODV acquires a route between the source-destination pair. However, in the case of DSDV, the route acquisition procedures are continually in effect by means of routing table exchanges between nodes, so that in principle, a connection request may be immediately serviced by routes maintained in the routing tables of source nodes. Yet, the dynamism in MANETs may lead to frequent invalidations of routes maintained by a proactive routing protocol so that a source node may be unaware of a route to a destination when a communication is sought. Then, the source node must wait until an update from the destination percolates through the network to the source and a route is known. Thus, proactive protocols may or may not suffer a route acquisition latency (excluding the propagation delay of the first packet) when faced with a connection request. Following on the above arguments, a definite route acquisition latency is associated with an inter-group connection made using GMRP; a latency may or may not be associated with route acquisitions for intra-group connections using GMRP.

The expanded ring-search mechanism of AODV allows for reducing the control overhead associated with a route request made to destinations located within a small hop distance from the source; intra-group connections can thus be serviced without network-wide flooding of route requests using AODV. However, in order to service connection requests between far-flung nodes (such as inter-group connections), AODV must repeatedly revise the extent of propagation of its RREQ packet so that it reaches a distantly located destination. Thus, inter-group connections may incur a large route acquisition delay when the expanded ring-search mechanism is in operation in AODV than otherwise. As for DSDV, the unsynchronized routing broadcasts made by nodes results in long propagation delays of routing updates made between distant nodes. Hence, inter-group communications are expected to suffer much longer route acquisition delays than AODV. The efficacy of the route request procedure in GMRP is contingent upon the infallibility of the proactive BN dissemination procedure. When an inter-group connection is sought using GMRP, the request is forwarded to all neighboring groups in the network; if the BN dissemination procedure fails to intimate source nodes of the current state of neighboring groups in the network, the semantics of the search may be disturbed. Hence, the timely dissemination of BN updates weighs tremendously on the inter-group route acquisition ability of GMRP. Further, the use of sub-optimal routes in GMRP's inter-group routing procedure will marginally prolong its computed route acquisition latency.

In Figure 5.11(a), a comparison of the number of fulfilled connections between the 3 proto-



(a) Number of Route Acquisitions

(b) Average Route Acquisition Delay

Figure 5.11: Comparison of Route Acquisition Characteristics under Different Traffic Categories

cols is made from 40 experiments conducted in this category. A total of 6000 connection requests were made of which 3000 were between group members and 3000 involved nodes belonging to different groups. Overall, it was observed that GMRP was able to fulfill 35% more connections than AODV and 45% more than DSDV. The number of intra-group connections that were established by AODV and GMRP was very similar. DSDV could however fulfill only 85% of that number of connections. In terms of inter-group connections, it is observed that GMRP secures thrice as many routes as AODV and about 2.5 times the number of connections fulfilled by DSDV. The fact that AODV fulfills the least number of inter-group connections amongst the 3 protocols is surprising. As much as route acquisition is a measure of the routing protocol's ability to reconcile to network dynamism, it is also partly circumstantial. For instance, node mobility could sometimes render a connection unavailable, so that a routing protocol may have to re-attempt route acquisition continuously until a connection is available again. It was observed that some of the connections secured by DSDV and GMRP which were not secured by AODV had inordinately long route acquisition delays (as high as 40 seconds). This indicates that these connections were unavailable when initially sought by the application and then became available later on when the relevant nodes lined up in a conducive fashion. The reason why AODV did not fulfill some of these connections may have been that the connections were unavailable at the time instants when AODV reattempted route requests for these connections[5]. The poor performance of DSDV can be traced to the enormous size of the routing updates from maintaining routes between all nodes in the network. Some of the routing information is never propagated because of the limited network bandwidth, resulting in the faliure of many connections. GMRP also relies on the periodic broadcast of routing information by nodes for intra-group routing; yet, routing updates are only as large as the group size. The fact that AODV fulfills very few inter-group connections can also be explained on the basis of the fairly heavy traffic load of 100 connections that are injected within a period of 50 seconds into the network. AODV must generate RREQ packets for each connection which may be potentially be required to be broadcasted throughout the network; the routing overhead generated is prohibitive and may have led to failure in the propagation of several RREQ packets.

Figure 5.11(b) compares the average route acquisition latencies of the 3 protocols. The average is computed based on the route acquisition latencies for those connections that were fulfilled by all 3 protocols. This is done to avoid disparities in the procurability and length between connections from disturbing a fair comparison of the route acquistion latency. Out of the total 6000 connections sought, only 1741 connections were successfully established by all 3 protocols. Of these, 477 connections were of the inter-group type while 1274 were of the intra-group type. The average route acquisition latency from 1741 connections for GMRP is 35% worse than that of AODV; DSDV performs 125% worse than AODV in this regard. In terms of route acquisition

[5] AODV attempts 3 back-to-back route requests followed by which it times out for 10 seconds waiting for a change in network state.

latency for intra-group connections, AODV incurs a 200% longer delay than GMRP and about 20% worse than DSDV. The delay values for intra-group group connections are evidently smaller than for inter-group connections; hence, the actual difference in route acquisition delay between GMRP and AODV is very little (about 0.08 seconds on the average). This difference is because AODV is a purely reactive protocol and must execute a route request procedure to establish each intra-group connection. On the other hand, GMRP and DSDV may possess advanced knowledge of routes to destinations (in some cases) from proactive route maintenance to these destinations. When considering inter-group connections, it is observed that GMRP's route acquisition latency is 75% worse than AODV, while DSDV performs 185% worse than AODV. The poor performance of GMRP with regards to inter-group connections can be pointed to the fact that it relies on a slow periodic broadcast-based BN dissemination scheme to infer the immediately surrounding topology of the group, and to a lesser extent, its use of sub-optimal routes.

CHAPTER SIX
CONCLUSIONS AND FUTURE WORK

The simulations in Experiment 5 confirm that group-based movement of nodes provides for a more favorable networking environment for routing protocols to function upon than nodal movement considered in conventional MANETs. Further, the entire set of simulations demonstrates that GMRP achieves a higher network throughput than both AODV and DSDV in representative cases of group mobility. The throughput-wise performance of AODV is observed to be marginally (about 5%) better than that of DSDV.

The expanded ring-search mechanism of AODV allows for the exploitation of spatial locality of connections when considering intra-group connections in group mobility scenarios. The network throughput of AODV is as high as DSDV for intra-group connection scenarios even under heavy loads for this reason. The expanded ring-search mechanism, however, increases the control overhead required to establish an inter-group connection since the ring radius for the RREQ packet must be revised over-and-over until the destination is located. AODV also crumbles when faced with heavy traffic loads because of the large volume of RREQs that are generated. Overall, AODV is observed to perform well with inter-group connections, and better than the other 2 protocols at low traffic loads; it is also able to negotiate large volumes of intra-group connections moderately well.

DSDV generates a large volume of control overhead from attempting to maintain routes to all network nodes. Further, the belated delivery of routing information to distant nodes renders DSDV less suitable to inter-group communication scenarios than AODV. The route acquisition delay of DSDV for inter-group communication scenarios in Experiment 6 is poorer than the other 2 protocols for the same reason. DSDV is, however, able to maintain intra-group routes reasonably well, despite the large overhead in from its routing updates. Yet, the network throughput achieved on intra-group connections by DSDV is much less than that of GMRP, and the total number of route acquisitions made by DSDV on intra-group connection scenarios in Figure 5.11(a)is only 85% that of GMRP or AODV. In summary, DSDV may be regarded as more competent with intra-group communication scenarios than with inter-group connections, and as immune to heavy traffic loads.

DSDV is generally unsuitable to large MANETs because of the enormousness of the routing tables from large node populations. Most of the MANETs studied in Experiments 1–6 were of large sizes, and yet, DSDV performs only 5% worse than AODV in a comparison of the achieved network throughput. This may be so from the reduction in the number of *triggered updates* sent by DSDV under group mobility. Group mobility lends more stability to the group topology; hence, the number of triggered updates made in the network is comparatively less than in a network where nodes move in an unrelated manner. This observation is also highlighted in Figures 5.9(a) and 5.10(a) where the network throughput achieved by DSDV sinks more rapidly than the other protocols with the addition of more individual nodes to the network. Thus, the underlying mechanisms of both AODV and DSDV, in their natural design, benefit from group mobility and the traffic model used on MANETs with groups, despite the lack of a notion of group-oriented behavior of nodes at the routing layer.

In GMRP, the size of periodic routing updates is constrained to a maximum of the group size, and the intra-group routing mechanism along with the BN dissemination scheme provide a structured abstraction of the network topology, upon which the inter-group routing scheme is overlaid. This structure forms the basis of the controlled flooding implemented in the route request procedure of the inter-group routing mechanism. The reduced reactive overhead is one reason why GMRP outperforms the other protocols even under heavy traffic loads. In fact the performance gain from employing GMRP increases with heavier traffic loads on the network. The peak performance of GMRP is also observed when the traffic on the network is intra-group biased. Yet, it is also verified that GMRP performs significantly better than both AODV and DSDV when the connection pairs are randomly chosen. Also, GMRP performs at least as well as AODV when the traffic consists purely of inter-group connections. Since the intra-group routing protocol has not been designed to any specific constraints in the relative mobility between group members, it is observed that node velocity and group mobility models do not affect the performance gain observed from employing GMRP. Experiment 3 suggests that the peak performance gain from using GMRP on MANETs that contain groups is observed when the network contains medium-sized groups. Lastly, it is observed that GMRP is able to adapt well to MANETs that contain a substantial percentage of individual nodes, wherein its performance is as good as AODV.

As mentioned earlier, the inter-group routing mechanism in GMRP is only pseudo-reactive: for its appropriate functioning, it needs the timely advertisement of BN information which is done in a proactive manner. An imminent RREQ broadcast in the inter-group routing module could be delayed by as long as the time it takes for a BN advertisement to penetrate to the peripheral nodes of a group, in the worst-case. This means that an route request can potentially be oblivious to certain *search directions* in the network when a BN advertisement is not delivered in time. The fact that the route request procedure hinges on a comparatively slower proactive BN advertisement is probably the reason why the route acquisition delay of GMRP for inter-group connections is 75% worse than AODV. It was also anticipated that GMRP would select longer routes for inter-group communication paths. As noted before, the average packet delay metric is more reflective of the route length used for communication than of the route acquisition latency or the time to recover from route failures. The measurements of the average packet delay for inter-group connection scenarios conducted in Experiments 1, 3 and 5 are fairly similar for all 3 protocols suggesting that, on the average, the inter-group routes chosen by GMRP are not drastically longer than that of DSDV or AODV. The average packet delay assessment must be interpreted with a little caution because different protocols deliver different number of packets, and, unlike the route acquisition delay, the average packet delay is not compared when all 3 protocols deliver a particular packet.

Overall, experimentation also suggests that the performance gains from GMRP are probably more significant in large-scale ad hoc networks which have heavier connection requirements

A performance comparison between GMRP and LANMAR has not been made because an ns-2 implementation of LANMAR is not available[6]. Given that LANMAR is a routing protocol that is specific tailored to the MANET environment that contains groups, a comparison of GMRP against LANMAR must be made to evaluate the differences between reactive and proactive routing strategies for inter-group communication.

The simulation of dynamism in group memberships has not been performed although the ns-2 implementation of GMRP is capable of inferring group splits and mergers. The simulation of dynamic groups must be performed with a two-fold objective. Firstly, it will help determine an empirical value for GRP_DISSOCIATION_PERIOD in Table 5.1, which is the time period for which a node will wait to hear of an intra-group routing update from another group member, failing which, it concludes the latter's dissociation from the group. A reasonable approximation of the GRP_DISSOCIATION_PERIOD is the time taken taken for a routing update to traverse the group diameter, that is, the time taken for a routing update to traverse the path between the farthest separated nodes in a group propagated according to the intra-group routing mechanism. Since this value depends on the size of a group, an empirical approximation is sought from the simulations of dynamic groups. A loss in network performance from GMRP's inability to establish changes in group memberships in a tmely manner is anticipated. The second objective of the simulation of

[6] Simulations of LANMAR in [12], [36] and [37] have been performed in GloMoSim [50].

dynamic group memberships is to revise the performance comparisons of GMRP against AODV and DSDV in these scenarios.

In Chapter 3, a service discovery-based approach to the formation of groups at the application layer and their configuration at the routing layer is proposed. If such an approach is employed, then the associated overhead from service advertisements/solicitations may not be factored into the routing layer since the mechanism aids group formation at the application layer itself. However, if such an approach were not to be used or if groups existed in a de facto manner, then a separate mechanism towards group configuration at the routing layer is necessary, whose overhead must be directly factored in the performance evaluation of GMRP. At any rate, the simulations have not addressed group formation and configuration; the implementation of these procedures, through a coordination with the application layer, and the revised performance evaluations of GMRP must be done. Further, the benefits of the service discovery approach to the application layer must be investigated, while possibly devising alternative means to the formation and sustenance of groups.

Appendix ONE
ROUTING ALGORITHM

/ The routing algorithm proposed for MANETs that demonstrate group mobility is presented here as a collection of event-driven procedures executed at each mobile node. An event is a stimulus such as the arrival of a packet at the routing layer or the expiry of a timer. When an event occurs, the routing layer is interrupted to handle the event. */*

```
If (Event == Received_Packet)                        /* When a packet is received. */
        {
        If (Packet_Type == Data_Packet)          /* Received packet is a data packet. */
                {
                If (Packet_Source == Self)              /* Data packet is received from a higher
                                                         layer on the same node. */
                        {
                        Look for route to destination in Intra-group routing table.

                        If valid route is available then dispatch packet to Next_Hop and
                        return.

                        Else, if route to destination is invalidated in routing table, queue
                        data packet and return.

                        Else, if destination does not belong in the same group,
                        then look up destination's GID in GID_Cache.
                        If not found, Goto label SEND_RREQ.

                        Look up best route (shortest number of groups) in Route_Cache to
                        destination's group.
                        If not available, Goto label SEND_RREQ.

                        Dispatch packet to the Next_Hop to reach Next_Group in determined route.
                        If no Next_Hop is currently available, then queue packet.
                        Return.

                        SEND_RREQ:
                        Check if RREQ has been dispatched for this destination before.

                        If yes, then queue data packet and return.
                        If not, create a RREQ packet for destination, and fill with sequence
                        number and other details.
                        Determine all Next_Hops to reach neighboring groups of self's
                        group.
                        Multicast RREQ to all Next_Hops.
                        Queue data packet and return.
                        }
                Else              /* I am either a forwarder or the destination of this data
                                     packet. */
                        {
                        If this is the destination, accept and pass packet to higher layers.
                        Return.

                        Else, is the incoming data packet an intra-group transmission?

                        If so, look for route to destination in Intra-group routing table.
                        If valid route is available then dispatch packet to Next_Hop.
                        Else, if route to destination is invalidated, or no entry for
                        destination exists in routing table, then queue data packet and
                        return.

                        If incoming data packet is of inter-group transmission type, then
                        look up source route in data packet.

                        Cache all routes and GID bindings in packet (or renew cache
                        timeouts), dispatch queued packets for any newly determined
                        routes.

                        If this is the last group specified in source route, determine
                        Next_Hop to destination from Intra-group routing table.
                        If route to destination is invalidated, queue packet and return.
                        Else, if route is available, then forward data packet and return.
                        If destination does not exist in routing table, drop packet, create
                        RERR packet indicating failed GID binding of destination.
                        Use source route of data packet to create forwarding route for
                        RERR back to data source.
                        Forward RERR and return.

                        If this is not the last group in source route of data packet, then
```

```
            determine Next_Group from source route.
            Determine Next_Hop to Next_Group, forward data packet and
            return.
            If Next_Group is unreachable (no node is a BN to Next_Group),
            then drop data packet and create RERR saying route has failed.
            Send RERR back to source and return.
            }
    }

If (Packet_Type == Intra-group_Routing_Update)        /* The received packet is an
                                                          intra-group routing update. */
    {
    Renew Neighbor_Timeout on corresponding neighbor.

    Handle intra-group routing update as per DSDV routing algorithm.
    Use BN information to populate BN tables.

    Use newly available routes to dispatch any queued packets.

    Invoke/revoke Group_Timeout for newly any validated/invalidated
    destinations.        /* Group_Timeout is described at the end of this Appendix. */
    Return
    }

If (Packet_Type == RREQ)        /* An inter-group Route Request has been received. */
    {
    Cache all routes and GID bindings in packet (or renew cache timeouts),
    dispatch queued packets for any newly determined routes.

    Can this packet be regarded duplicate (check sequence number and
    Reject field) or is its TTL value equal to 0?

    If so, drop packet and return.

    If not, add record of RREQ to RREQ_Cache.
    Check Intra-group routing table, GID_Cache and Route_Cache for
    route to destination.
    If route is available, create RREP and forward to source based on reverse
    route in RREQ. Discard RREQ and return.
    If no route is available, am I the GL of my group?
```

```
            If I am not, forward RREQ to GL and return.
            If I am GL, multicast RREQ to all neighboring groups except ones already
            traversed by the RREQ, and return.
            }

        If ( Packet_Type == RREP)           /* A Route Reply message has been received. */
            {
            Cache all routes and GID bindings in packet (or renew cache timeouts),
            dispatch queued packets for any newly determined routes.

            If I am destination of RREP, drop packet and return.
            Else, forward RREP to destination and return.
            }

        If ( Packet_Type == RERR)           /* A Route Error message has been received. */
            {
            Cache all routes and GID bindings in packet (or renew cache timeouts),
            dispatch queued packets for any newly determined routes.

            Extract failed route and GID information from RERR and delete self's
            cache information accordingly.

            Lookup Failed_Cache to see if equivalent RERR was previously received.
            If so, discard packet and return.

            Else, make note of this RERR in Failed_Cache.
            If I am not destination of RERR, forward RERR to destination and return.
            If I am destination of RERR, send out new RREQ to initially-sought
            destination with failed information. Drop RERR and return.
            }

        }

    If (Event == Routing_Broadcast_Timeout)        /* An intra-group routing broadcast has been
                                                      scheduled. */
        {
        Send intra-group routing broadcast according to DSDV, including recent changes
        in BN information.
        Return.
        }
```

```
If (Event == Reply_Timeout)                /* A Route Reply has not yet been received for a
                                           particular Route Request. */
        {
        Send fresh RREQ with higher sequence number to timed-out destination.
        Return.
        }

If (Event == Cache_Timeout)          /* A cache timeout has been triggered resulting in
                the removal of the corresponding entry from the cache. 4 types of cached
                information exists: Route_Cache that holds all inter-group routes known,
                GID_Cache that holds all Node-GID bindings known, RREQ_Cache that
                remembers the RREQ packets that have lately been serviced at the node, and a
                Failed_Cache that contains information pertaining to the dispatch of RERR
                messages to source nodes whose inter-group routes have broken down. */
        {
        Delete corresponding cache entry. (Route_Cache/GID_Cache/Failed_Cache/RREQ_Cache).
        Return.
        }

If (Event == Neighbor_Timeout)               /* A routing update has not been received from a
                                           neighbor for a certain time period. */
        {
        Remove timed-out node from Neighbor_List.
        Invalidate associated routes according to DSDV.
        Invoke Group_Timeout on routes invalidated.
        Return.
        }

If (Event == Group_Timeout)                /* A route to an intra-group member has not been
                                      available for a certain time period pointing to a possible
                                      disaffiliation of the node from the group. */
        {
        Remove entry for timed-out node from intra-group routing table.
                                           /* Destination is assumed to have left group. */
        Return.
        }
```

BIBLIOGRAPHY

[1] C.E. Perkins and E.M. Royer. Ad hoc on-demand Distance Vector routing. In *Proceedings of IEEE WMCSA 99*, pages 90–100, Feb. 1999.

[2] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced Distance-Vector routing (DSDV) for Mobile Computers. *ACM SIGCOMM Computer Communication Review*, 24(4), Oct 1994.

[3] Bangnan Xu, S. Hischke, and B. Walke. The role of ad hoc networking in future wireless communications. In *Proceedings of International Conference on Communication Technology (ICCT)*, volume 2, pages 1353–1358, April 2003.

[4] A. Acharya, A. Misra, and S. Bansal. MACA-P: A MAC for concurrent transmissions in multi-hop wireless networks. In *Proceedings of First IEEE International Conference Pervasive Computing and Communications*, pages 505–508, March 2003.

[5] J. Monks, V. Bharghavan, and W. Hwu. A power controlled multiple access protocol for wireless packets networks. In *Proceedings of IEEE INFOCOM 01*, volume 1, pages 219–228, April 2001.

[6] W. Hung, K. Law, and A. Leon-Garcia. A dynamic multi-channel MAC for Ad Hoc LAN. In *Proceedings of 21st Biennial Symposium on Communications*, pages 31–35, June 2002.

[7] N. Jain, S. Das, and A. Nasipuri. A multichannel CSMA MAC protocol with receiver-based channel selection for multihop wireless networks. In *Proceedings of the IEEE ICCCN 01*, pages 432–439, Oct 2001.

[8] D.B. Johnson and D.A. Malt. Dynamic Source Routing in Ad Hoc Wireless Networks. *Chapter 5, Mobile Computing*, 1996. Kluwer Publishing Company.

[9] V.D. Park and M.S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proceedings of IEEE INFOCOM 97*, pages 1405–1413, April 1997.

[10] Y.B. Ko and N.H. Vaidya. Location-aided routing (LAR) in mobile Ad Hoc networks. In *Proceedings of ACM/IEEE MOBICOM 98*, pages 66–75, Oct. 1998.

[11] S. Basagni, I. Chlamtac, V.R. Syrotiuk, and B.A. Woodward. A distance routing effect algorithm for mobility (DREAM). In *Proceedings of ACM/IEEE MOBICOM 98*, 1998.

[12] Guangyu Pei, Mario Gerla, and Xiaoyan Hong. LANMAR: Landmark routing for Large Scale wireless Ad Hoc networks with group mobility. In *First Annual Workshop on Mobile and Ad Hoc Networking and Computing, MobiHOC '00.*

[13] G. Pei, M. Gerla, and T.-W. Chen. Fisheye state routing in mobile Ad Hoc networks. In *Proceedings of the 2000 ICDCS Workshops*, pages D71–D78, April 2000.

[14] G. Pei, M. Gerla, X. Hong, and C.C. Chiang. A wireless hierarchical routing protocol with group mobility. In *Proceedings of IEEE WCNC 99*, pages 1536–1540, Sep. 1999.

[15] A. Bruce McDonald and Taieb Znati. A dual-hybrid adaptive routing strategy for wireless ad-hoc networks. *Wireless Communications and Networking Conference*, 3:1125–1130, Sep. 2000.

[16] Xiaoyan Hong, Mario Gerla, Guangyu Pei, and Ching-Chuan Chiang. A group mobility model for ad hoc wireless networks. In *Proceedings of 2nd ACM international workshop on Modeling, analysis and simulation of mobile and wireless systems, MSWiM 99*, 1999.

[17] Fan Bai, Narayanan Sadagopan, and Ahmed Helmy. IMPORTANT: A framework to systematically analyze the Impact of Mobility on Performance Of RouTing protocols for Ad hoc NeTworks. In *Proceedings of IEEE INFOCOM*, pages 825–835, April 2003.

[18] Narayanan Sadagopan, Fan Bai, Bhaskar Krishnamachari, and Ahmed Helmy. PATHS: Analysis of PATH duration Statistics and their impact on reactive MANET routing protocols. In *Proceedings of the 4th ACM international symposium on Mobile Ad hoc Networking and Computing MobiHoc '03*, 2003.

[19] J. M. Jaffe and F.H. Moss. A responsive distributed routing algorithm for computer networks. *IEEE Transactions on Communications, COM*, pages 1758–1762, July 1982.

[20] Shree Murthy and J. J. Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. *Mobile Networks and Applications*, 1(2):183–197, Oct 1996. Kluwer Academic Publishers.

[21] T. Clausen and P. Jacquet et. al. Optimized link state routing protocol, Oct. 2003. RFC 3626 Available at `://www.faqs.org/rfcs/rfc3626.html/`.

[22] B. Bellur and R.G. Ogier. A reliable, efficient topology broadcast protocol for dynamic networks. In *Proceedings of IEEE INFOCOM '99*, volume 1, pages 178–186, March 1999.

[23] Zygmunt J. Haas. A new routing protocol for the reconfigurable wireless networks. In *6th International Conference on Universal Personal Communications Record, 1997*, volume 2, pages 562–566, Oct 1997.

[24] M. R. Pearlman and Z. J. Haas. Determining the optimal configuration for the zone routing protocol. *IEEE Journal on Selected Areas of Communications*, 17:1395–1414, Aug. 1999.

[25] Z.J. Haas and M.R. Pearlman. The performance of query control schemes for the zone routing protocol. *ACM/IEEE Transactions on Networking*, 9(4):427–438, 2001.

[26] Prince Samar, Marc R. Pearlman, and Zygmunt J. Haas. Independent zone routing: An adaptive hybrid routing framework for ad hoc wireless networks. *ACM/IEEE Transactions on Networking*, 12(4), Aug. 2004.

[27]] Lan Wang and Stephan Olariu. A two-zone hybrid routing protocol for mobile ad hoc networks. *IEEE Transactions On Parallel And Distributed Systems*, 15(12), Dec. 2004.

[28] C. Santivanez, R. Ramanathan, and I. Stavrakakis. Making link-state routing scale for Ad Hoc networks. In *Proc. of MOBIHOC '01*, Oct 2001.

[29] Venugopalan Ramasubramanian, Zygmunt J. Haas, and Emin Gun Sirer. HARP a Hybrid Adaptive Routing Protocol for mobile Ad Hoc networks. In *Proc. of MobiHoc 03*, June 2003.

[30] Atsushi Iwata, Ching-Chuan Chiang, Guangyu Pei, Mario Gerla, and Tsu-Wei Chen. Scalable routing strategies for Ad Hoc wireless networks. *IEEE Journal On Selected Areas In Communications*, 17(8), Aug. 1999.

[31] Yaacov Fernandess and Dahlia Malkhi. K-clustering in wireless ad hoc networks. In *Proc. of POMC 02*, Oct. 2002.

[32] D.J. Baker and A. Ephremides. A distributed algorithm for organizing mobile radio telecommunication networks. In *Proceedings of the 2nd International Conference on Distributed Computer Systems*, pages 476–483, April 1981.

[33] Foroohar Foroozan and Kemal Tepe. A high performance cluster-based broadcasting algorithm for wireless ad hoc networks based on a novel gateway selection approach. In *Proc. of PE-WASUN 05*, Oct. 2005.

[34] P. Krishna, N.H. Vaidya, M.C hatterjee, and D.K. Pradhan. A Cluster-based approach for routing in Dynamic Networks. In *Proceedings of the Second USENIX Symposium on Mobile and Location-Independent Computing, 1995.*

[35] Ying Ge, L. Lamont, and L. Villasenor. Hierarchical OLSR - a scalable proactive routing protocol for heterogeneous Ad Hoc networks. In *Proc. of IEEE International Conference on Wireless And Mobile Computing, Networking And Communications, WiMob '05*, volume 3, pages 17–23, Aug. 2005.

[36] M. Gerla, X. Hong, and G. Pei. Landmark routing for large ad hoc wireless networks. In *Proceedings of IEEE GLOBECOM 2000*, Nov. 2000.

[37] Xiaoyan Hong, N. Nguyen, Shaorong Liu, and Ying Teng. Dynamic group support in LANMAR routing ad hoc networks. In *4th International Workshop on Mobile and Wireless Communications Network*, pages 304–308, Sep. 2002.

[38] Xiaoyan Hong, M. Gerla, and Li Ma. Multiple-landmark routing for large groups in ad hoc networks. In *Proceedings of MILCOM 2002*, volume 1, pages 495–500, Oct. 2002.

[39] Ken Blakely and Bruce Lowekamp. A Structured Group Mobility Model for the simulation of Ad hoc networks. In *Proceedings of the second international workshop on Mobility management and wireless access protocols, MobiWac 04*.

[40] Mirco Musolesi, Stephen Hailes, and Cecilia Mascolo. An Ad Hoc mobility model founded on social network theory. In *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems, MSWiM '04*, Oct. 2004.

[41] Z.R. Zaidi, B.L. Mark, and R.K. Thomas. A two-tier representation of node mobility in ad hoc networks. In *First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, IEEE SECON '04*, pages 153–161, Oct. 2004.

[42] J.C. Cano, P. Manzoni, and M. Sanchez. Evaluating the impact of group mobility on the performance of mobile ad hoc networks. In *IEEE International Conference on Communications*, volume 7, pages 4039–4043, June 2004.

[43] J.M. Ng and Yan Zhang. Impact of group mobility on ad hoc networks routing protocols. In *The 8th International Conference Advanced Communication Technology, ICACT 2006*, volume 2, Feb. 2006.

[44] S. Nesargi and R. Prakash. MANETconf: Configuration of Hosts in a Mobile Ad Hoc Network. In *Proceedings of InfoCom' 02*, June 2002.

[45] Mansoor Mohsin and Ravi Prakash. IP Address Assignment in Mobile Ad Hoc Networks. In *Proceedings of MILCOM'02*.

[46] C.E. Perkins, J.T. Malinen, R. Wakikawa, E.M. Belding-Royer, and Y. Sun. IP Address Autoconfiguration for Ad Hoc Networks, July 2000. Published as draft-ietfmanet-autoconf-01.txt by IETF, MANET Working Group.

[47] Yih-Chun Hu and David B. Johnson. Ensuring Cache Freshness in On-Demand Ad Hoc Network Routing Protocols. In *Proceedings of POMC '02*, Oct. 2002.

[48] Yih-Chun Hu and David B. Johnson. Caching Strategies in On-demand Routing Protocols for Wireless Ad Hoc Networks. In *Proceeding of MobiCom '00*, 2000.

[49] The Network Simulator-NS-2, 2004. http://www.isi.edu/nsnam/ns.

[50] M.Takai, L.Bajaj, R.Ahuja, R.Bagrodia, and M.Gerla. GlomMoSim: A Scalable Network Simulation Environment. Technical Report 990027, UCLA, Computer Science Department, 1999.