AN ABSTRACT OF THE THESIS OF

Ho Sung Kang for the degree of Master of Science in Electrical and Computer Engineering presented on June 3, 2009.

Title: Urban Military Operation Mobility Model

Abstract approved: ____
Ben Lee

A Mobile Ad-hoc NETwork (MANET) is collection of wireless mobile nodes without a network infrastructure or centralized administration. Although MANETs can be used in many applications, such as mobile Internet, military communication, and disaster relief networks, a number of challenges remain. These include routing, medium access control, security, scalability, energy efficiency, mobility, etc. In order to study the viability of large-scale MANETs, researchers rely on wireless network simulators to test new ideas. Wireless network simulations require several important parameters, such as routing protocols, mobility models, and data traffic models. Among these, developing realistic mobility models is crucial for accurately evaluating the performance of MANETs.

There are many models that emulate mobility of users. The most representative are *entity* and *group mobility* models. In a battlefield, mobility patterns of military units are different than mobility patterns of civilians. Thus, a special group mobility model is needed to appropriately simulate military operations on the battlefield. Hong *et al.* proposed the Reference Point Group Mobility (RPGM) [1] model, which relates a group movement by a logical center. The Virtual Track (VT) [2] mobility model and Reference Region Group Mobility (RRGM) [3-5] model adds a group partitioning and merging scheme. The VT mobility model uses a Switch Station for group partitioning and merging, and the RRGM

employs the Reference Region for assigning a group mission and dividing a group. These mobility models can be used to model operations in open areas or within specific building structures. However, group movements for military operations exhibit a more complicated pattern in urban areas. For example, a group is divided into smaller groups for accomplishing new mission and the small groups must be merged with their main forces at specific location or group destination after achieving their new tasks. Unfortunately, VT and RRGM mobility models do not specify a group merging event and location with their main forces.

This thesis proposes a new group mobility model for military urban operation called *Urban Military Operation Mobility Model* (UMOMM). In UMOMM, the group moves along a road and employs group partitioning for new missions and allows for merging at the specific locations. UMOMM also employs a time delay to model soldiers encountering and overcoming obstacles during a missions. Finally, the impact of the proposed mobility model on different routing protocols is studied.

Urban Military Operation Mobility Model

by
Ho Sung Kang

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented June 3, 2009
Commencement June 2009

Master of Science thesis of Ho Sung Kang presented on June 3, 2009

APPROVED:

Major Professor, representing Electrical and Computer Engineering

---

Director of the School of Electrical Engineering and Computer Science

---

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Ho Sung Kang, Author

ACKNOWLEDGEMENTS

I would like to express my gratitude to my major professor, Ben Lee, for his encouragement and inspiration throughout my study. The accomplishment of this thesis would not have been possible without Dr. Lee's constant guidance. And I also want to express appreciation to Professor Huaping Liu, Roger Traylor, and Mike Pavol for their serving in my graduate committee.

I would like to thank to Korean Army and Defense Security Command for giving chance to study and fully support in OSU.

Finally, I would like to extend my sincere thanks to my family, my mother, Soonim Kim, have been encouraging for whole my life, my wife, Jinae, for her support, love, and friendship throughout our years, and my lovely children, Soomin and Jimin.

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (Continued)

Page

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF TABLES

1. Introduction

The flow of information is one of the most important issues for humans because the quality of decisions we make depends on the accuracy of information. In addition, information transfer has evolved from word of mouth to mobile networks, and the amount of information and speed of distribution have increased dramatically in past decades. Timely and accurate dissemination of information is important in all facets of life. But, another very important use of information is in military activities. Military network communication focuses on how to link nodes effectively and transfer data fast and accurately.

Military wireless communications started with simple Morse code to send messages for *Command and Control* (C2). Recently, *Mobile Ad-hoc NETworks* (MANETs) have been employed as military wireless networks because they can operate without a preplaced infrastructure [6]. Even now, the main part of military wireless network communications supports only C2 for military operations. However, because of the advances in network technology and increase in the amount of information, military networks play a central role in *Network Centric Warfare* (NCW). NCW is a new theory that focuses on increasing combat power by using effective linking or networking among resources of military forces [7, 8]. It enables the sharing of battlefield awareness by self-synchronization and other network-centric operations to achieve commanders' intent.

In NCW, we can imagine soldiers wearing small display monitors to observe the real-time battlefield situation around them. Moreover, they would also wear a small helmet-mounted video camera to report real-time visual information, which would increase the accuracy of information and aid in decision making for commanders. In order for such high-tech soldiers to become a reality, the problem of handling increased network traffic must be solved. Thus, most research on military wireless networks has been performed on MANETs because military units cannot assume the existence of a network infrastructure in an area of operation.

In MANETs, the network topology frequently changes based on the movement of mobile nodes. Therefore, *routing protocols* are crucial for maintaining some degree of connectivity, even as nodes move. There are a number of routing protocols for MANETs, which include proactive and reactive routing protocols [9]. However, the performance of routing protocols has been difficult to evaluate because of lack of mobility models that realistically represent the behavior of mobile nodes.

Mobility models can be classified into two types; trace-based and synthetic mobility models [10, 11]. *Trace-based* mobility models are based on observing movements of real-life systems or robots that perform a common task. The accuracy of trace-based models depends on the number of participants and the time of observation. *Synthetic* mobility models imitate realistic behavior of mobile nodes using random and probabilistic processes. Although trace-based models are more accurate than synthetic models, trace-based models are difficult to obtain. This is because trace data could include private information, which restricts its collection and distribution, and is difficult to model even after the traces are collected [12]. That is why most mobility models are based on the synthetic model.

There are many mobility models based on the synthetic model for MANETs, as well as ad hoc networks. In general, synthetic mobility models can be classified into two groups: entity and group mobility models. In an *entity mobility model,* each node moves independently with its own destination and velocity. They include the Random Waypoint mobility model [13] and Random Direction mobility model [14]. These mobility models are widely used to analyze the performance of routing protocols. However, in a military operation scenario, nodes are not always independent. Mobility among nodes is related to each other. In a *group mobility model*, mobile nodes belong to a group and they usually move together to the same destination. Moreover, their movements are influenced by not only the group members but also nearby groups.

A number of group mobility models have been suggested and implemented. These include the Reference Point Group Mobility (RPGM) model, Reference Velocity Group Mobility (RVGM) model[15], Diamond Group Mobility (DGM) model [16], and Structured Group Mobility Model (SGMM) [17]. In these models, a mobile node is permanently affiliated with a pre-defined group and the group mobility pattern is fixed to the reference node or a group leader's movement.

In real combat operations, however, there are more complex mobility scenarios depending on the situation and military units. One typical characteristic is that a group can dynamically partition itself into a number of subgroups or merge with another group. For instance, in military operations in urban areas, a number of army units will first mobilize outside the urban area. When operation orders are given, the units will move toward their destinations within the urban environment. During the operation, a group may be divided into several subgroups where some of the subgroups are assigned new tasks while the rest of the subgroups continue towards their original objectives. After completing the new mission, a subgroup will rejoin its main force.

There are mobility models that support group partitioning and merging scenarios, which include the Virtual Track (VT) model [2] and Reference Region Mobility Model

(RRGM) [3]. However, the VT model and RRGM do not specify the relationship and command structure among groups. Therefore, they are not sufficient for modeling mobility patterns of soldiers in urban areas under dynamically changing situations.

Therefore, this thesis presents a group mobility model, called *Urban Military Operation Mobility Model* (UMOMM). In UMOMM, a group can be dynamically partitioned for new tasks and be merged with its main force at an arbitrary point on the group's route or destination. Moreover, UMOMM can model situations where each group encounters various obstacles constructed by hostile forces and must overcome them to reach the destination.

The thesis is organized as follows: Section 2 presents a background on MANET routing protocols. Section 3 overviews mobility models, including entity and group mobility models. Section 4 presents the proposed UMOMM. Section 5 presents the simulation environment and compares the performance results of UMOMM for three routing protocols against RWP and RPGM. Finally, Section 6 concludes the thesis and discusses future work.

2. MANET Routing Protocols

MANETs are different from other wireless networks such as infrastructure mode Wireless Local Area Networks (WLANs) and Metropolitan Area Networks (MANs). In MANETs, there are no access points or base stations for mobile stations to communicate with. Moreover, routing paths can dynamically change due to mobility. Therefore, most MANET routing protocols concentrate on how to establish the best routing paths in order to increase throughput. This section presents proactive and reactive routing protocols for MANETs.

2.1. Proactive Routing Protocols

In proactive routing protocols, each node examines routes to all the nodes within the network and maintains an up-to-date routing table. Routing information is either updated at regular intervals or when the network topology changes. Thus, a proactive routing protocol is also called a "table-driven" routing protocol [9]. Two popular proactive routing protocols for MANETs are the *Destination-Sequenced Distance Vector* (DSDV) protocol [18], which is based on the distributed Bellman-Ford algorithm, and the *Optimized Link State Routing* (OLSR) protocol [19].

2.1.1. Distributed Bellman-Ford (DBF) algorithm

The Bellman-Ford algorithm finds the shortest path using the link cost between source and destination nodes. In the Bellman-Ford algorithm, the shortest path is not based on hop counts but measured by total link costs. For example, between nodes *A* and *D* in Figure 1, there are four available paths; $A \rightarrow B \rightarrow D$, $A \rightarrow B \rightarrow E \rightarrow D$, $A \rightarrow C \rightarrow E \rightarrow D$, and $A \rightarrow C \rightarrow E \rightarrow B \rightarrow D$. When hop count is considered, the path $A \rightarrow B \rightarrow D$ (2 hops) is the best. However, when total link cost is considered, the path $A \rightarrow C \rightarrow E \rightarrow D$ (3 hops) is the minimum link cost.

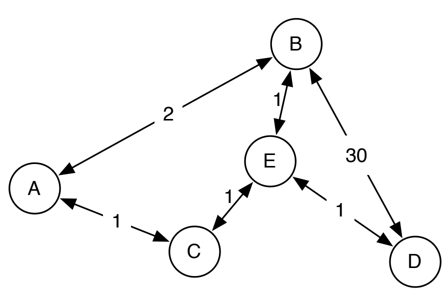
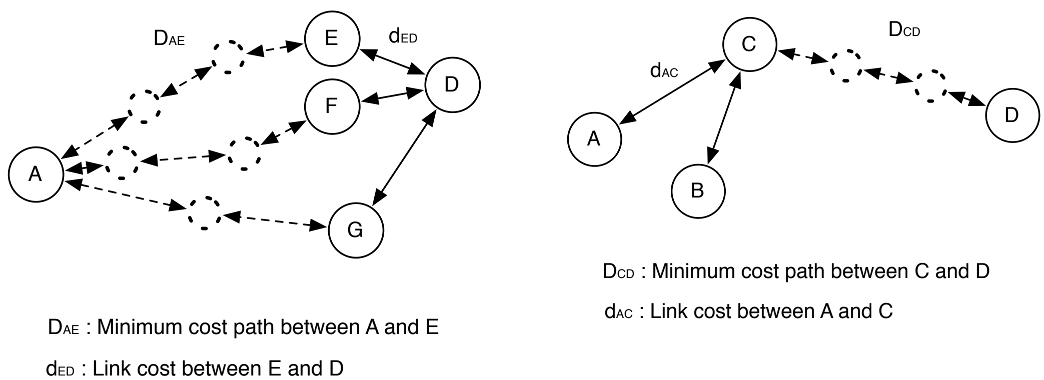


Figure 1: Example of five nodes network with link cost

There are two approaches for finding the shortest path between two nodes: centralized and distributed. The centralized method computes the cost of the shortest path to all the nodes prior to the destination. For example, as illustrated in Figure 2(a), node *A* computes the cost of the shortest path from *A* to *D*. The distributed method minimizes the computations required to determine the shortest path. For example, as illustrated in Figure 2(b), if node *A* needs the shortest path between nodes *A* and *D*, node *A* receives the minimum cost information between nodes *C* and *D* from its neighbor node *C*. Similarly, if node *B* needs the minimum cost path between nodes *B* and *D*, node *B* receives node *C*'s shortest path to *D* [20]. Therefore, the distributed Bellman-Ford algorithm is suitable for distributed MANETs.

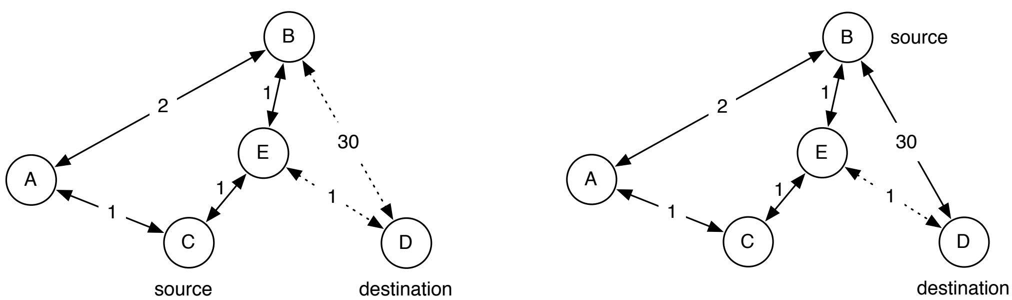


(a) Centralized scheme (b) Distributed scheme

Figure 2: Centralized and distributed Bellman Ford algorithm [20]

There are two disadvantages in the Bellman-Ford algorithm, which are the loop and count-to-infinity problems. Figure 3(a) illustrates the loop problem, where the three links *B*→*D*, *B*→*E*, and *E*→*D* have become disconnected. Node *E* broadcasts its own routing table containing the incorrect information of the *B*→*D* connection, and node *B* also broadcasts its own routing table containing the incorrect information of the *E*→D connection. Nodes *A* and *C* update their routing tables based on *B*'s or *E*'s routing table. When a transmission is needed between nodes *C* and *D*, node *C* sends the data to node *E*, and node *E*, which is located outside of node *D*'s transmission range, sends back the data with a recommendation to use node *B*. Node *C* tries to send the data to node *B*, and node *B*, being also located outside of node *D*'s transmission range, sends back the data, including a recommendation to use node E. The data from source node *C* repeatedly travels the route *C*→*E*→*C*→*A*→*B*→*A*→*C*.

Figure 3(b) illustrates the count-to-infinity problem, where the link between nodes *E* and *D* has become broken. Node *B* tries to send data to node *D* using the route *B*→*E*→*D*. However, node *E* sends data back to *B* because there is no path between nodes *D* and *E*. Node *B* updates its routing table, which increases the link cost between nodes *E* and *D* by one. Although the link cost between nodes *E* and *D* is increased, the total cost of the route *B*→*E*→*D* is still lower than the cost of the route *B*→*D*. Therefore, node *B* repeatedly sends the data to node *E* and updates the routing table until the link cost exceeds 30. Then, node *B* switches to the route *B*→*D*.



(a) The loop problem

(b) The count-to-infinity problem

Figure 3: Disadvantages of DBF

2.1.2. Destination-Sequenced Distance Vector (DSDV)

In DSDV, information contained in the routing tables is transmitted among the nodes and is updated at each node. The routing tables include the list of all reachable destination nodes as in the Bellman-Ford algorithm. Moreover, DSDV adds a sequence number to each route, originated by the destination node, indicating how old the route is. To keep routing tables up-to-date, each node either periodically sends its routing table or immediately sends its routing table when a link is broken or a new link is added. When a node receives several route updates from different sources, old routes are updated only if the sequence number of the received routing update is higher. The addition of sequence numbers eliminates the loop and count to infinity problems [18].

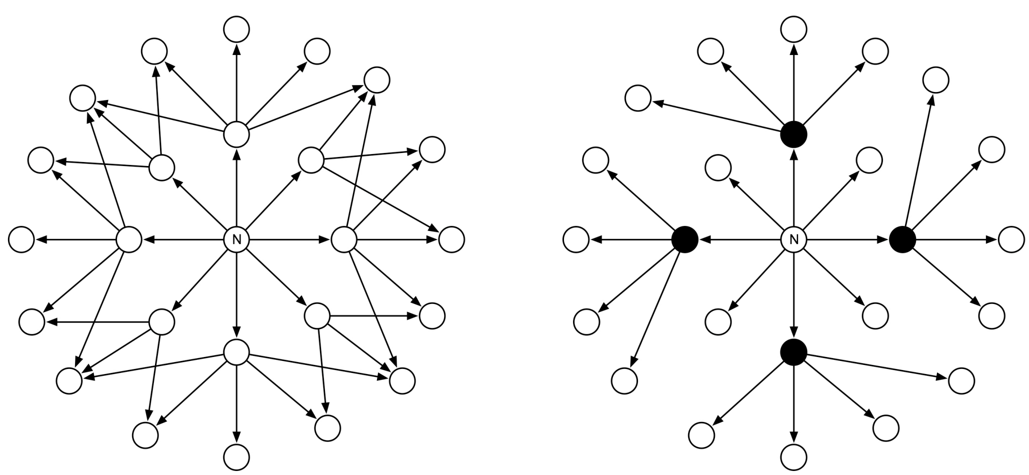
As mentioned before, the DSDV protocol regularly updates the routing tables, which increases the number of route maintenance packets and decreases bandwidth efficiency. Moreover, when the network topology changes, a new sequence number must be created. This procedure increases network delay. Therefore, this protocol is more suitable for low mobility networks.

2.1.3. Optimized Link State Routing (OLSR)

The OLSR protocol is another proactive link-state routing protocol for MANETs [19]. The OLSR is designed to reduce routing messages among the nodes. This is achieved using Multipoint Relays (MPRs), which are the only nodes that are allowed to retransmit link-state updates.

Two control messages are used to select MPRs and exchange link-state information; HELLO and Traffic Control (TC) messages. HELLO messages are periodically broadcasted to all 1-hop neighbor nodes. This allows each node to establish connections with its immediate neighbor nodes and also exchange information with nodes that are 2 hops away for selecting MPRs. The TC messages are also periodically broadcasted by MPRs to maintain topological information about the network [21].

The selection of MPRs is done by having each node select a MPR set from among its neighbor nodes and all nodes that are 2 hops away (called the minimized number of MPR set). For example, Figure 4(a) shows node *N* with 8 neighbor nodes and 16 2-hop nodes. Node *N* first exchanges HELLO messages with 1-hop nodes and then identifies all the 2-hop nodes. After calculating the minimizing MPR set from 1-hop and 2-hop node information, node *N* divides neighbor nodes into four MPR sets and four neighbor nodes as shown in Figure 4(b). The four MPR sets can retransmit network topology information using TC messages, while the four neighbor nodes can only receive and update their own link-state information



(a) (b)

Figure 4: An example of MPR selection [21]

The routing table of each node can keep an up-to-date the list of available destinations because of proactive routing protocol. If a node receives a control message notifying of a change in network topology, the node recalculates the routing path using the shortest path algorithm.

OLSR can adapt to dynamic movements by adjusting the interval for TC messages. The interval is increased for a low mobility or stable network; otherwise, the interval is decreased. Thus, a dense network is the best environment for OLSR [22].

Since OLSR maintains all available destinations in its routing table, like DSDV, the routing overhead also increases as the number of nodes increases.

2.2. Reactive Routing Protocols

In a reactive routing protocol, routing paths are not maintained regularly and are sought only when a source node needs them. Thus, reactive routing protocols are called "on-demand" routing protocols [9]. There are two main operations in reactive routing protocols: route discovery and route maintenance. If a node or mobile station has data to transmit to a destination node, it starts the *route discovery* procedure to find the appropriate routing paths. The *route maintenance* procedure is started when the active routing path is disconnected because of node mobility or a device is turned off. Since the routing tables are changed only when the source node needs to transmit the data, the route maintenance is an important process for reactive routing protocols.

Since reactive routing protocols decrease the routing overhead, they are suitable for large and high mobility networks. Furthermore, these protocols use sequence numbers to eliminate the loop problem. However, these protocols suffer from increased delay.

2.2.1. Dynamic Source Routing (DSR)

The DSR [23] protocol is simple, self-configuring and self-organizing, based on the route discovery and maintenance mechanism. Therefore, there is no centralized control and it is suitable for high mobility networks.

In order to establish a communication between nodes *A* and *E* shown in Figure 5, node *A* initiates the route discovery procedure. Node *A* broadcasts a Route Request (RREQ) packet to all the connected nodes with a unique request identification (ID number), which is determined by Node *A*. The RREQ also contains a "hop limit' to restrict the number of intermediate nodes the RREQ can travel. After node B receives the RREQ, it checks its routing table to find the active routing path. If node *B* has an active routing path, it sends this information back to node *A* using a Route Reply (RREP). If node *B* does not have a routing path, node *B* initiates the routing discovery procedure. This process repeats until the RREQ reaches node *E*, where node *E* responds with a RREP. The RREP packet travels in the reverse direction and all nodes are updated with the new routing path. If a link is broken during the sending of packets or the updating of the routing table, the route maintenance begins. If node *A* has alternative routes, the routing path is changed; otherwise, the route discovery procedure is initiated.

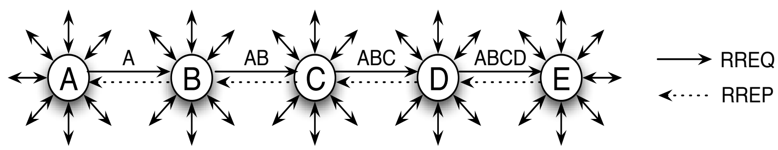


Figure 5: DSR route discovery procedure

There are a few additional features to improve performance. First, the routing table is added only in the forward link direction to prevent appending of useless routing information called "caching overheard routing information". Second, a problem known as route reply storm may occur. This is illustrated in Figure 6 where, after neighbor nodes receive a RREQ from node *A*, each node responds with a RREP, based on its own routing table. Therefore, node *A* receives the routing path from neighbor nodes at the same time, which could cause collisions and create a short delay. To prevent route reply storm, each node waits for a random amount of time before sending a RREP [6].

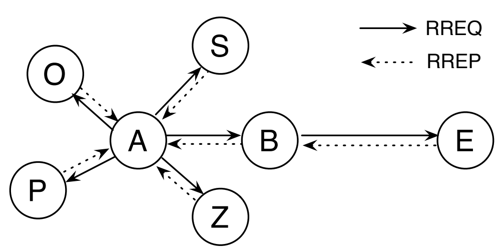


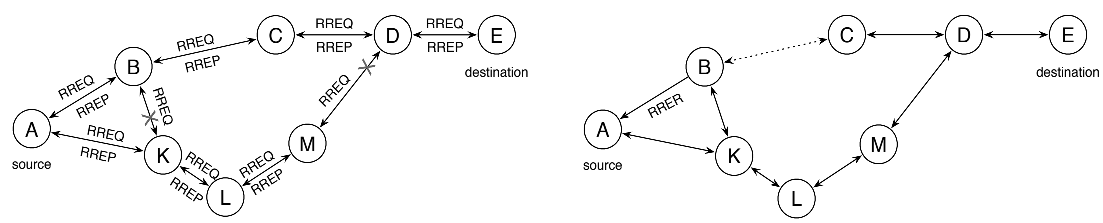
Figure 6: Route reply storm

2.2.2. Ad hoc On-demand Distance Vector (AODV)

AODV [24] is capable of both unicast and multicast routing. It uses routing discovery and maintenance methods similar to DSR's to generate and maintain the routing tables. The

RREQ of the AODV protocol includes the sequence number, IP address of source and destination nodes, and broadcast ID.

For example, as shown in Figure 7(a), when a node *A* initiates a RREQ to find destination node *E*, if an intermediate node has the routing path, it responds with a RREP based on its own routing table. Otherwise, node *E* responds with the RREP. Afterwards, node *A* establishes the routing table *A*→*B*→*C*→*D*→*E*. AODV reduces the control overhead by using broadcast ID. As illustrated in Figure 7(a), when node *K* receives a RREQ from both nodes *A* and *B* with the same broadcast ID, node *K* which already received the RREQ from node *A* checks the broadcast ID and deletes the RREQ packet from node *B*. To reduce propagation of useless RREQ packets, a source node sets the Time To Live (TTL), which is referred to as the "ring search" method.



(a) Route discovery

(b) Route maintenance

Figure 7: Route discovery and maintenance

The route maintenance procedure is started when a link is broken or the network topology is changed. For example, Figure 7(b) illustrates a broken link between nodes *B* and *C*. When node *B* recognizes the broken link, it transmits a Route Error (RERR) message to the source node. After node *A* receives the RERR, it deletes the active routing path and broadcasts the RERR packet to neighbor nodes and reinitiates the route discovery procedure. Finally, node *A* reestablishes the new routing path *A*→*K*→*L*→*M*→*D*→*E*.

AODV uses HELLO messages to inform that a node is alive to its neighbor nodes. To prevent useless broadcasting of HELLO messages, the TTL of a HELLO message is set to one. When a node receives a HELLO message, it updates the lifetime of neighbor nodes in its routing table. If a node does not receive a HELLO message for a predefined time, a node recognizes the broken link and deletes the link from its routing table. This will generate RERR messages when other nodes try to use this broken link.

AODV also eliminates the loop and count-to-infinity problems by employing sequence numbers.

3. Mobility Model

In MANETs, mobile nodes dynamically create or change the network topology by their movements and thus routing paths are needed to exchange data among the nodes. Recently, many researchers have proposed new routing protocols for not only maintaining connectivity but also efficient packet delivery. Network simulation is very important for evaluating the performance of routing protocols in MANET, and one of the most widely used network simulators is *Network Simulator 2* (NS2) [25]. In NS2, or any wireless network simulator for that matter, a mobility model must be predefined. In this section, two types of synthetic mobility models are discussed: entity mobility models and group mobility models.

3.1. Entity Mobility Model

In this subsection, four entity mobility models are presented that define moving patterns of individual mobile nodes. Random WayPoint (RWP) is the most common and simple mobility model for evaluating routing protocols for MANETs. In the presence of geographical constraints where mobility is limited to streets or highways, the Manhattan, Freeway, and Graph-based Mobility Models are closely related to RWP.

3.1.1. Random Waypoint

The RWP mobility model was proposed by Johnson and Maltz [13], and is widely implemented in ad hoc network simulations because of its simplicity. NS2 also includes a tool called *setdest* for generating the random waypoint model.

RWP has three main parameters; maximum velocity (V_{max}), pause times (T_{pause}), and destination. At the beginning of a simulation, each mobile node randomly selects the source and destination waypoints within the simulation area, which are independent of those selected by other mobile nodes. Each mobile node moves towards its destination with a constant velocity taken from [0, V_{max}]. On arriving at the destination, a mobile node remains stationary for a predefined pause time defined by T_{pause}. After a duration of T_{pause}, the node moves to the next target waypoint with a steady velocity selected from [0, V_{max}]. The process of selecting a destination, a velocity, and a movement is repeated until the simulation ends. Figure 8 shows an example movement trace of three nodes.

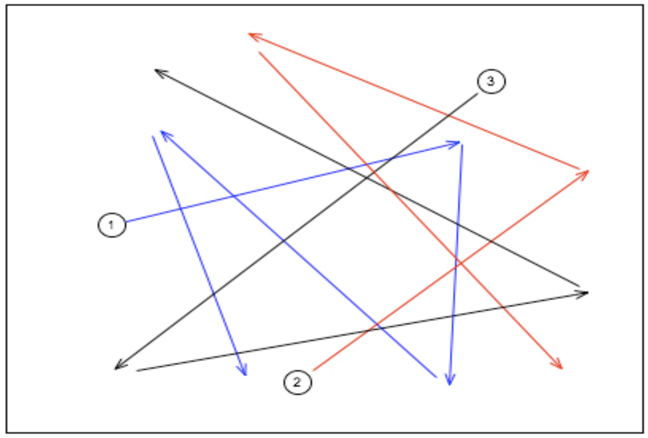


Figure 8: Random WayPoint

The RWP model can be used to simulate various mobility situations. For example, a high mobility scenario, such as vehicle mobility, can be simulated by simply increasing V_{max} and decreasing T_{pause}. On the other hand, pedestrian mobility can be modeled by decreasing V_{max} and increasing T_{pause}. As a result, many ad hoc network routing protocols are evaluated using RWP.

However, as mobile nodes repeat their movement, the node density tends to be very high at the center of the simulation area, whereas the node density is almost zero at the borders. This phenomenon called *border effect* or *density wave* was first observed by Bettstetter [26] and Royer et al. [14]. Figure 9 illustrates the node density on a 1000m × 1000m simulation area.

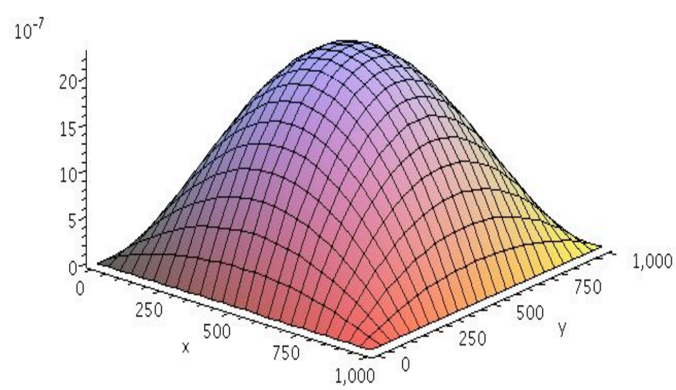
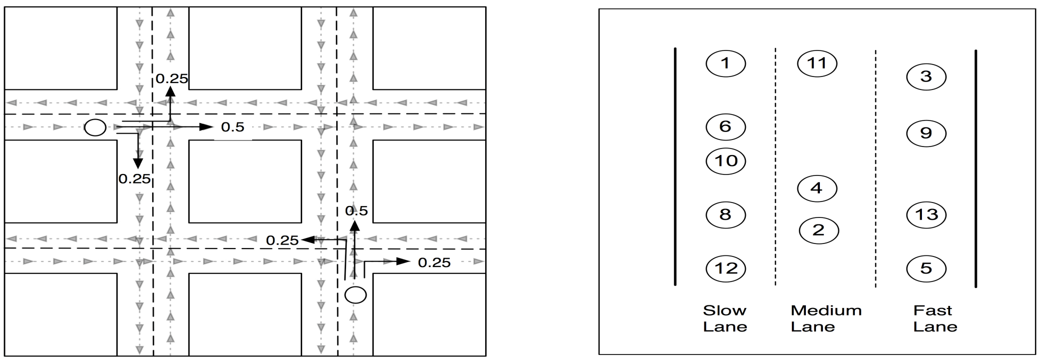


Figure 9: Node density of RWP (1000m x 1000m area) [27]

3.1.2. Manhattan and Freeway Mobility Model

Bai *et al.* introduced the Manhattan Mobility model and the Freeway Mobility model for simulating a metropolitan area [28]. In the Manhattan Mobility model, all node movements are specified on the streets defined by a map. This model is useful for modeling movements in an urban area. The map is composed of horizontal and vertical streets, each having two lanes. Upon reaching an intersection, the mobile node decides its direction using probabilistic methods, e.g., probabilities of going straight, turning left, and turning right are 0.5, 0.25, and 0.25, respectively, as illustrated in Figure 10(a).



(a) Manhattan mobility model (b) Freeway mobility model

Figure 10: Manhattan and Freeway mobility model [28]

The Freeway Mobility model also uses a map composed of freeways of several lanes in both directions. Each mobile node is restricted to its lane on the freeway. Furthermore, if two mobile nodes, e.g., nodes 3 and 9 in Figure 10(b), travelling in the same direction come within the Safe Distance (SD), the velocity of the following node 9 cannot exceed the velocity of node 3. Although movement patterns of mobile nodes using the entity model are independent, mobile nodes in the freeway model are influenced by other mobile nodes travelling in the same lane, and vice versa.

3.1.3. Graph-based Mobility Model

As the name suggests, a graph is used to represent mobility restricted by buildings and streets. The graph is composed of vertices and edges. The vertices are the locations where mobile nodes might visit and the edges between these locations represent streets or train connections. An example of graph-based modeling of a city center is shown in Figure 11.

Initially, the mobile node randomly selects the first source and the destination on the graph. The mobile node moves from its initial location to its destination along the shortest possible path. After arriving at the destination, it stays for short time and picks the next destination on the graph. This process is repeated until the simulation ends.

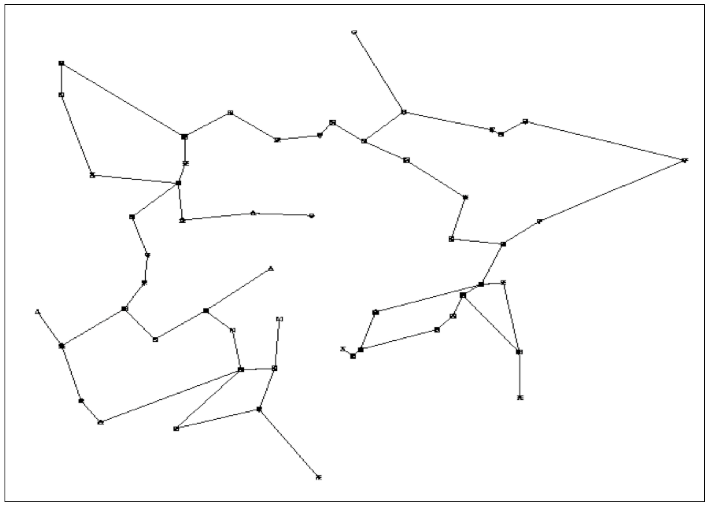


Figure 11: An example of a graph-based model [29]

The speed of the mobile node can be selected between v_{min} and v_{max} and the pause time at each destination can be between $t_{staymin}$ and $t_{staymax}$. For example, a typical pedestrian speed is defined as v_{min} = 2km/h and v_{max} = 5km/h, while people visiting at a shop or train station is defined as $t_{staymin}$ = 120s and $t_{staymax}$ = 600s [30].

3.2. Group Mobility Model

In MANETs, unlike the entity mobility model, there are many situations where mobile nodes move together. Therefore, these mobile nodes become a group whether or not they create a specific formation. For example, military units are composed of a large number of soldiers with a hierarchical command structure. Each group is assigned tasks, such as attacking the enemy, occupying an area, and rescuing civilians or friendly forces. All group members work together in a collaborative way to achieve their mission. Firefighters and relief teams in a disaster area are also examples of group collaboration [31].

This section discusses four group mobility models that imitate such collaboration. The Reference Point Group Mobility Model (RPGM) is the general group mobility model. Others, such as the Structured Group Mobility Model (SGMM), the Virtual Track (VT) mobility model and the Reference Region Mobility Model (RRGM), are extensions of RPGM. In particular, the VT mobility model and RRGM employ a group partitioning and merging scheme.

3.2.1. Reference Point Group Mobility (RPGM)

RPGM takes into consideration the spatial relationship and movement behavior among the members of a group [1]. Each group is composed of a logical center, called the group leader, and a number of group members. The logical center determines the entire group's motion behavior, including location, speed, direction, acceleration, and reference point. The other nodes, called group members, are randomly distributed around the reference point. The motion behavior of group members is decided by randomly deviating from their group leader. RPGM can be used to represent various military units (e.g., infantry, artillery, and armor) during military operations and rescue teams (e.g., firemen, policemen and medical assistants) in disaster relief efforts.

As illustrated in Figure 12, the group leader moves from current group area $X(t)$ to next group area $X(t+1)$ by group motion vector $\overrightarrow{GM}^{t}$, which is randomly chosen by its own simulation scenario. Upon reaching the destination, the group leader determines a new group motion vector $\overrightarrow{GM}^{t+1}$ and moves again from $X(t+1)$ to $X(t+2)$.

The group motion vector, $\overrightarrow{GM}^{t+n}$, also provides movement parameters to group members. Initially, each member is located in the neighborhood of the group leader. Figure 12 illustrates an example of determining members' motion vectors, which are decided by deviating from their group's motion vector by some degree within the group area. The member motion vector $\overrightarrow{MM}_i^{t}$ of node *I* at time *t* is given as follows:

$$\overrightarrow{MM}_i^{t} = \overrightarrow{GM}^{t} + \overrightarrow{RM}_i^{t} \qquad (3\text{-}1)$$

The random motion vector $\overrightarrow{RM}_i^{t}$ of each node is indirectly assigned by movement behavior from its group leader [32].

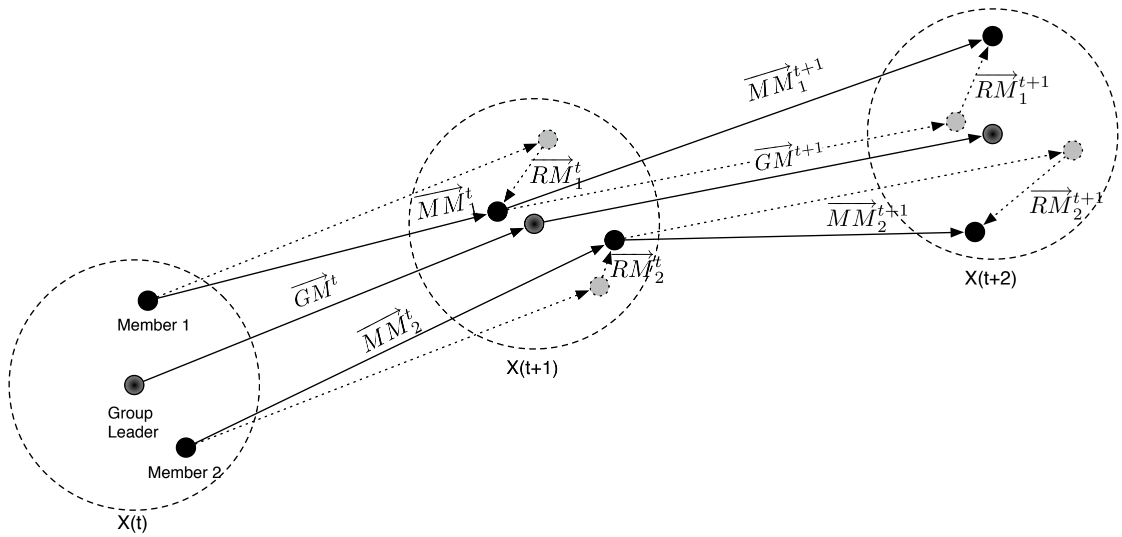


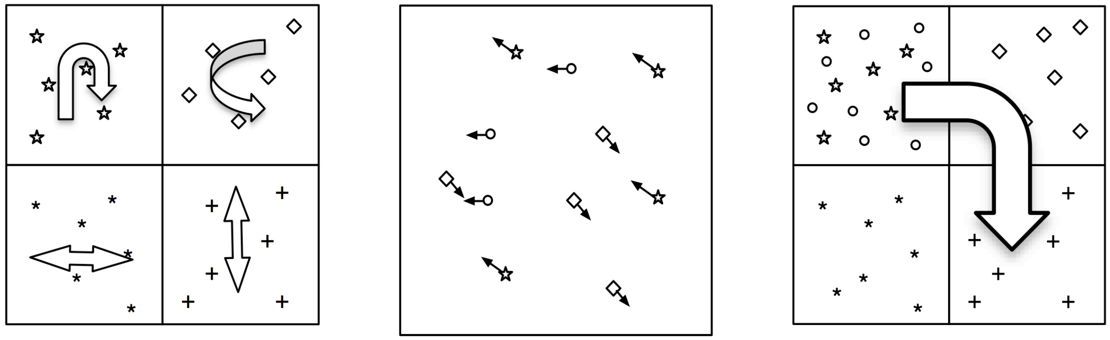
Figure 12: Reference Point Group Mobility (RPGM)

With proper selection of node mobility parameters in RPGM, various scenarios of group mobility can be simulated. Three group mobility scenarios are presented as follows [1]:

(i) *In-place mobility model* - the entire simulation area is divided into smaller sub-areas. Then, each group is located in a single sub-area as shown in Figure 13(a). This scenario can be used for military operations, where several units are executing the same operation in different areas. Large-scale disaster relief operations can also be implemented using this model.

(ii) *Overlap mobility model* - unlike the in-place mobility model, different groups with different tasks carry out their mission in the same place, as shown in Figure 13(b). For example, in a disaster area, each group (e.g., firemen, policemen, and medical assistant team) performs its own mission in the same area.

(iii) *Convention mobility model* - the simulation area is divided into several sub-areas and some groups are initially located in one of the sub-areas. These groups are allowed to visit different areas, as shown in Figure 13(c). For example, in a convention, there are several exhibition rooms and a group of participants can travel to different rooms.



(a) In-place model (b) Overlap model (c) Convention model

Figure 13: Three scenarios of RPGM [1]

3.2.2. Structured Group Mobility Model (SGMM)

SGMM is an extension of RPGM [17]. As shown in Figure 14, a group j has the reference point c_j, called the geographical center of the group. To maintain movement towards the destination, a c_j has a directional orientation of angle T on a global coordinate system. The locations of group members or subordinate groups are determined by an angle and a distance relative to c_j. The node i occupies a place, which is determined by a distance d_i from c_j from a given distribution D, and an angle a_i away from T from a given distribution A. This way, a group can keep its desired structure whether it moves or not. Since all the positions of nodes or subgroups depend on their c_j, the movement of c_j controls all the mobile nodes in the simulation. Thus, this model does not need velocity factors of individual nodes.

SGMM can handle multiple group simulations by applying the model recursively. If the structure of groups has a hierarchical organization, the leader group plays the role of the reference group of the SGMM. The reference point of each group is related to the reference point of the leader group and all group members of a group. This way, not only the structure of a single group is maintained but also the formation of multiple groups.

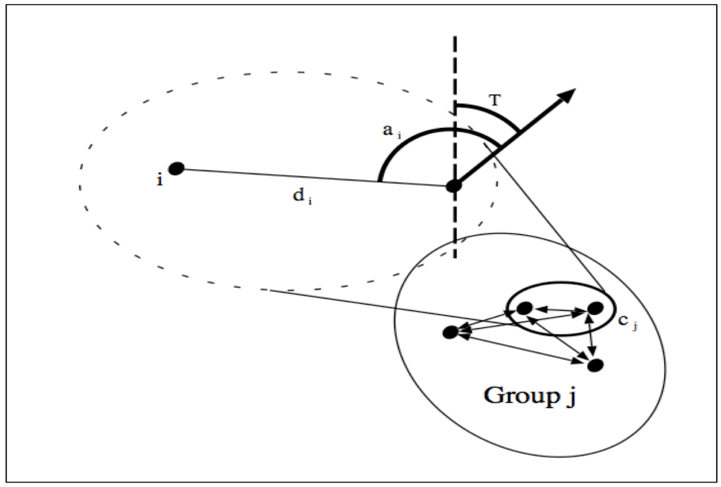
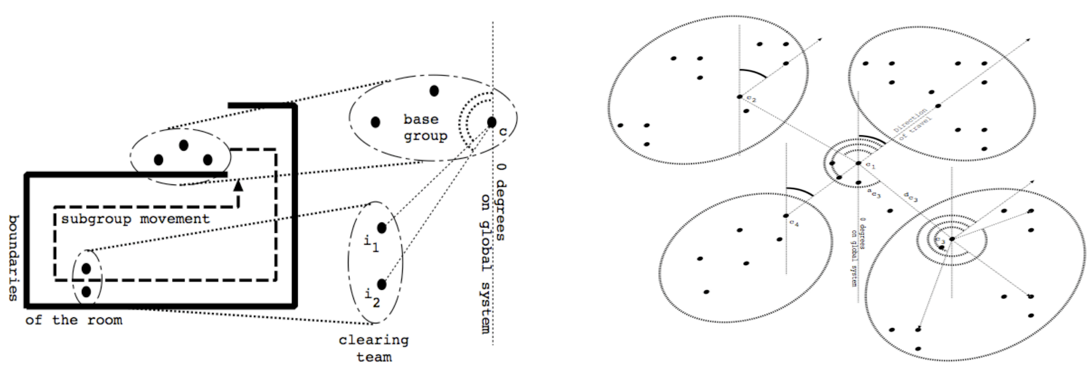


Figure 14: Structured Group Mobility Model [17]

SGMM can be applied to various situations as described below [17]:

(i) *Firefighters operating in a building* - Firefighters carry out their tasks in several groups. The group structure and control is critical when they attack the fire or rescue victims in the building. Figure 15(a) shows the operation of locating a fire or searching for victims in the room. In this operation, the command elements are at the entrance of the room and smaller search teams move through the room.

(ii) Military units on the battlefield - The military unit in a hierarchical formation moves toward its destination. Figure 15(b) illustrates a tank battalion consisting of several tank units moving in a structure formation in an open area.



(a) Firefighting team in building (b) Military unit including subgroup

Figure 15: Application of SGMM [17]

3.2.3. Virtual Track (VT) Mobility Model

The VT mobility model defines a simulation area as switch stations and virtual tracks, and models group partitioning and merging [2]. The switch stations are randomly deployed in the simulation area and connected by virtual tracks. The number of switch stations and the maximum length of virtual tracks are defined by the user.

The groups are distributed along the virtual tracks and individual nodes are distributed within the simulation area. The groups must use the virtual tracks for their movement but the individual nodes can move anywhere, as in the RWP model. A group selects a switch station for its destination and moves toward it. Upon reaching the destination, switch station, each node checks its stability value. If the stability value is beyond the group stability threshold value, this node can select a new destination that is different from its group. A group partitioning occurs when nodes of the group choose a different switch station. Group merging occurs when several groups, which arrive at the same switch station, decide on the same virtual track to the next destination. Figure 16 illustrates group partitioning and switch stations.

The group movement is based on the RWP model with one constraint: The intermediate point must be closer to the destination than the previous point and be in the same virtual track

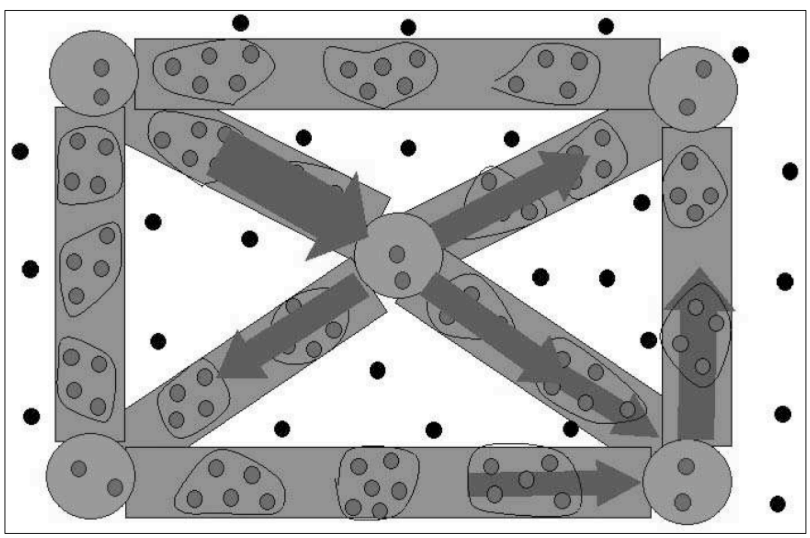


Figure 16: Virtual Track based mobility model [2]

3.2.4. Reference Region Group Mobility (RRGM)

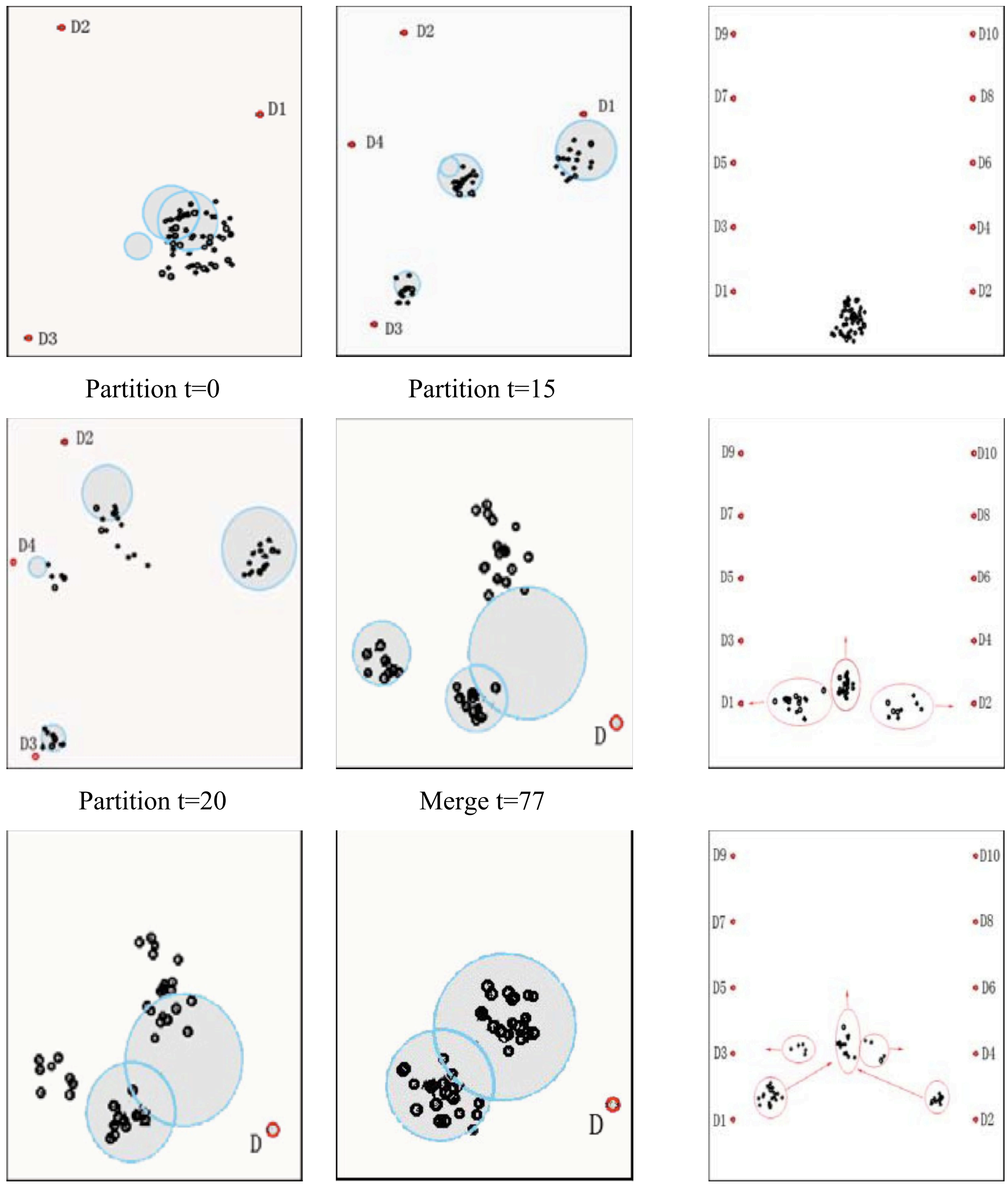
The RRGM model is based on group partitioning and merging for more realistic group movement patterns [3]. In this model, there are several groups and every group is associated with a reference region. A reference region is an intermediate location or destination where the group moves to and the group's scope of activity depends on node density. The sites of the reference region define the intermediate points where a group will move to on its way to the destination. Upon arriving at a reference region, nodes will randomly move around within the region while waiting for other nodes. After waiting for a while at an intermediate region, nodes move toward the next area.

In RRGM, if several reference regions are assigned to one group, this group will be partitioned into a number of subgroups associated with different destinations if sufficient nodes are available. When a subgroup reaches the destination, it could merge with another group. Groups are categorized into two types: active and standby groups. Active groups work with new reference regions, moving towards the destination and moving around within the region, whereas standby groups do not have a new reference region and just move around their own region and wait for a new assignment.

Two scenarios are suggested for simulation implementation [3]:

(i) Search and rescue model - A group is assigned a new task, which could be composed of one or several reference regions. The group is separated into a number of subgroups equal to the number of reference regions if nodes are available. After completing their mission, these subgroups merge with other groups or carry out another mission as shown in Figure 17(a). Rescue teams and military operations are good examples of the search and rescue model.

(ii) Room searching or exhibition hall visiting model - In the case of a building search, as shown Figure 17(b), policemen form a team and move along the corridor and split into subgroups to search rooms. After searching a room, the subgroups rejoin the main group and move to next room. Group partitioning and rejoining is repeated until the mission is completed. As another example, a group of people enters an exhibition hall, and some people may pass by the exhibition counter while others visit the counter. Afterwards, they will rejoin the main group.



Partition t=0 Partition t=15

Partition t=20 Merge t=77

Merge t=82 Merge t=85

(a) Search and rescue model

(b) Room searching and exhibition hall visiting model

Figure 17: Scenarios of RRGM [3]

4. The Proposed Mobility Model

In this chapter, a new group partitioning and merging mobility model called *Urban Military Operation Mobility Model* (UMOMM) is proposed. UMOMM is based on the group mobility model with restrictions in movements in urban areas. In the proposed model, a platoon represents a basic group. A platoon consists of 27 soldiers organized into three squads consisting of one main squad and two subordinate squads. Each squad consists of one leader and 8 member nodes. A platoon is associated with intermediate locations, where it must visit and stop for a random time, and a destination.

The simulation area is divided into a border area and an urban area, as shown in Figure 18. The border area is the outskirts of the urban area where mobile nodes are initialized and the urban area is composed of vertices and edges like a Graph-based mobility model. The vertices are locations where mobile nodes might visit and the edges are connections, e.g., streets, between these vertices. In the UMOMM model, the vertices represent the main locations of platoon activities, such as beginning and end points, intermediary stopping points, as well as positions for group partition and merging.

The platoon leader, who is the leader of the main squad, determines the shortest path route from the starting to the destination point via the intermediate locations. Then, the platoon moves to the next vertex along the predefined route. After reaching the next vertex, the platoon leader stops for a period of time and determines the partition of squads. The platoon leader also determines the merging point located on the route and the direction that the partitioned squads move toward. After partitioning, the platoon leader moves toward the next vertex on the route, and the partitioned squads move in their new directions and calculate the shortest path to the merge point. At the merge point, the squads merge and then follow the pre-defined route of the platoon leader. The process of stopping, partitioning, and merging will repeat until the platoon reaches its destination.

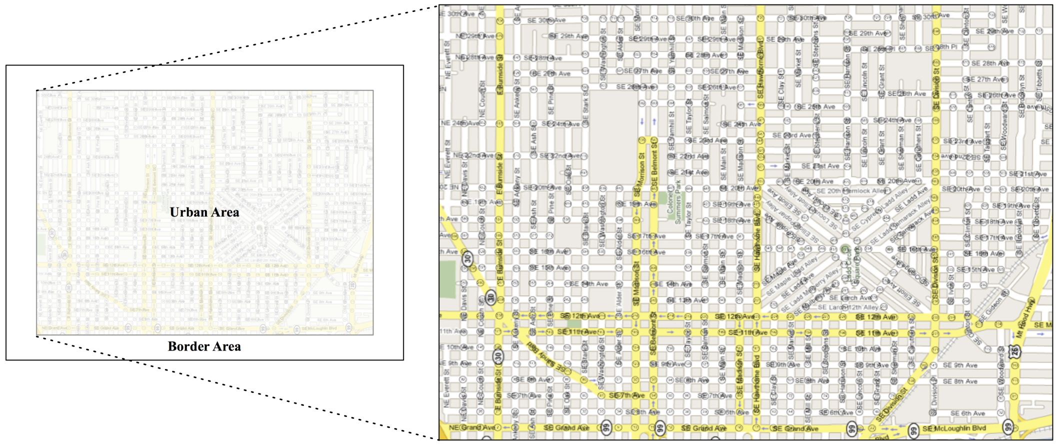


Figure 18: Simulation area

4.1. Model Initialization

The initial location of a platoon is selected based on the tactical plan of the operation. The leader of the first subordinate squad is then located a distance D_1 and angle θ_1 from the platoon leader, where D_1 is randomly selected between D_{max} and D_{min} and θ_1 is randomly chosen between 90 and 180 degrees. The location of the second subordinate squad's leader is similarly selected based on distance D_2 and angle θ_2 from the platoon leader, where D_2 is randomly selected between D_{max} and D_{min} and θ_2 is randomly chosen between -90 and -180 degrees. Figure 19 illustrates the positioning of two squad leaders relative to the platoon leader.

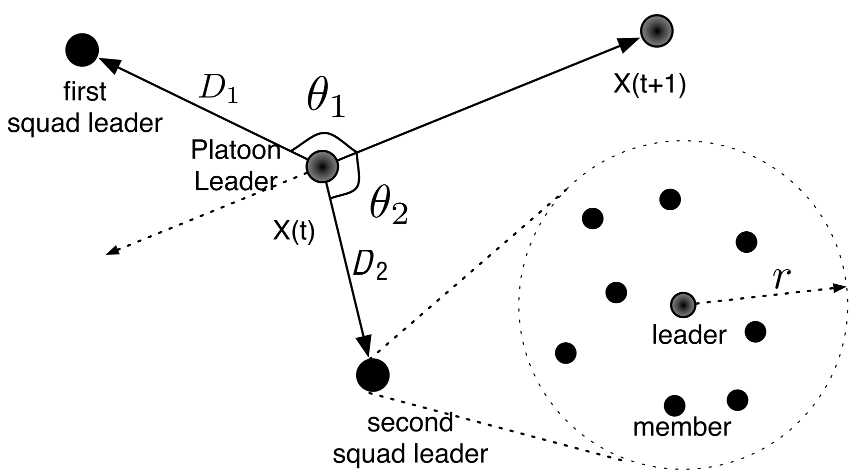


Figure 19: First location of group

Once the positions of squad leaders are determined, the member nodes of each squad are randomly distributed within a radius *r* from their squad leader.

4.2. Node Movement

The platoon leader decides the shortest path, which is a series of vertices to the destination via the intermediate locations. The subordinate squads are initially associated with the platoon leader's route. Once the route is set, the main squad starts moving to the first vertex on the route. After waiting for some time, the first subordinate squad starts following the platoon leader. Similarly, the second subordinate squad waits for same time after the first subordinate squad moves.

Each squad forms two columns as it maneuvers. Figure 20 shows an example where the two columns march on each side of the road. The distance *d* between two columns is related to the width of the road. The spacing *d'* among the mobile nodes within a column is determined by the equation:

$$d'_i = D_0 \times \gamma_i \tag{4-1}$$

where D_0 is the initial distance, for each mobile, and γ_i is a random value in the range of (0.5, 1). The distance of the individual node is maintained until the squad reaches the next vertex.

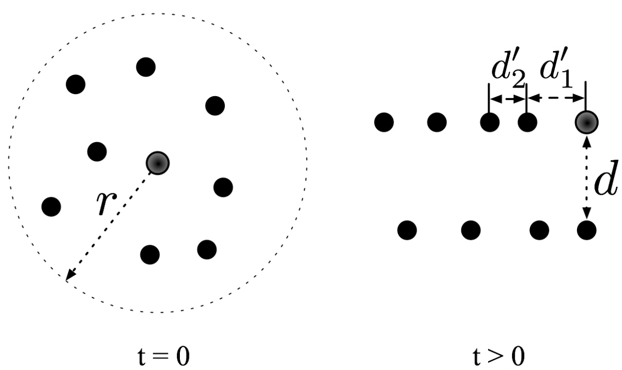


Figure 20: Formation of group movement

The velocity of the soldiers is the normal speed of infantry movement [33]. Upon reaching the next vertex, the group stays for a period of time to decide whether or not to partition or check its route and recalculate the individual distance d'_I before moving towards the next vertex on the route. This process is repeated until the destination is reached.

4.3. Group Partitioning and Merging

In military operations, a group has to be split into a number of small subgroups to carry out new missions in different areas. For example, a platoon moving toward its destination could be divided into several squads to reconnoiter or gain control of the area adjacent to the main path of troops. Furthermore, the partitioned subgroups must be merged with the main group after accomplishing their tasks.

How the group partitioning and merging is performed is not well defined and depends on the situation.

4.3.1. Group Partitioning

In UMOMM, it is necessary to check the availability of squads, since group partitioning can occur at every vertex the group visits. If the squad was already partitioned before arriving at the current location, the platoon leader just stays for some time and moves on to the next vertex. Otherwise, in order to select accessible vertices among vertices connected to the current vertex, the platoon leader calculates the angle between the axis from the current location to the destination and all the connected vertices from the current position. After computing all the adjacent vertices, the platoon leader selects accessible routes whose angles are between -90 and 90 degrees, excluding its own of travel route. For example, Figure 21 illustrates the process of partitioning into subgroups. As can be seen, there are five paths, $\vec{x}_0$ to $\vec{x}_4$, from the current position. After calculating angles of all the paths, only two routes, $\vec{x}_1$ and $\vec{x}_4$, whose angles are between -90 and 90 degrees, can be selected to partition the route. $\vec{x}_0$ is excluded because it represents the route of the group leader.

The group partitioning is performed for several different scenarios based on the number of squads and accessible vertices. First, if a platoon has only one squad and only one accessible vertex, the platoon leader simply matches the squad to the route. Second, if there is one squad and more than one accessible path, the platoon leader randomly selects one of the accessible paths for the squad. Third, if there are two squads and one partitioning path, then the platoon leader randomly chooses one of the squads for the route. Fourth, if there are two squads and more than two paths, the platoon leader first decides on a number of partitioning squads and then one of the paths is randomly chosen. Afterwards, partitioning scheme defaults to the first case.

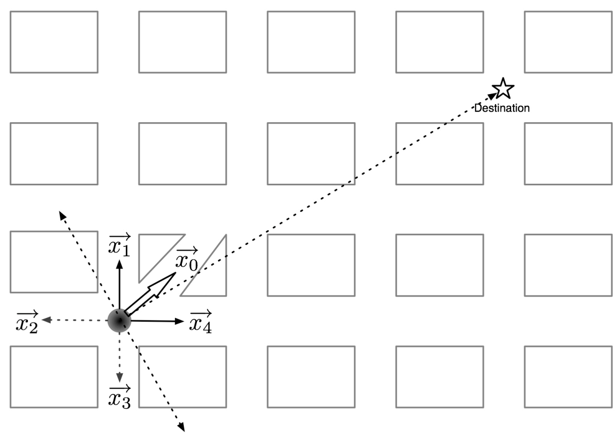


Figure 21: Group partitioning

4.3.2. Group Merging

After the group is partitioned, the platoon leader also determines the merging point, which is an arbitrary point on the platoon leader's route, and designates that location to each partitioned subgroup. When the partitioned squad moves along the new route and arrives at the next vertex, the leader of the squad calculates its new path, which is the shortest path to reach the merging point. If the next vertex is a dead end, the partitioned squad turns back to the previous vertex and calculates the shortest path to the merging point. When a platoon leader or subordinate squad arrives at a merging point, it must wait for the rest of the group. After merging, the subordinate squad follows the platoon leader until a new group partitioning occurs.

4.4. Blocking Area

In urban military operations, troops encounter *blocking areas* (BAs), such as destroyed roads and/or obstacles constructed by hostile forces. BAs cause different delay times for the moving groups. For example, in a military operation, an ambush requires more time to overcome than removing a barricade. Therefore, BAs are categorized into several levels depending on the severity of the situation, called blocking levels. Based on this, the delay time T_d for a BA is defined as

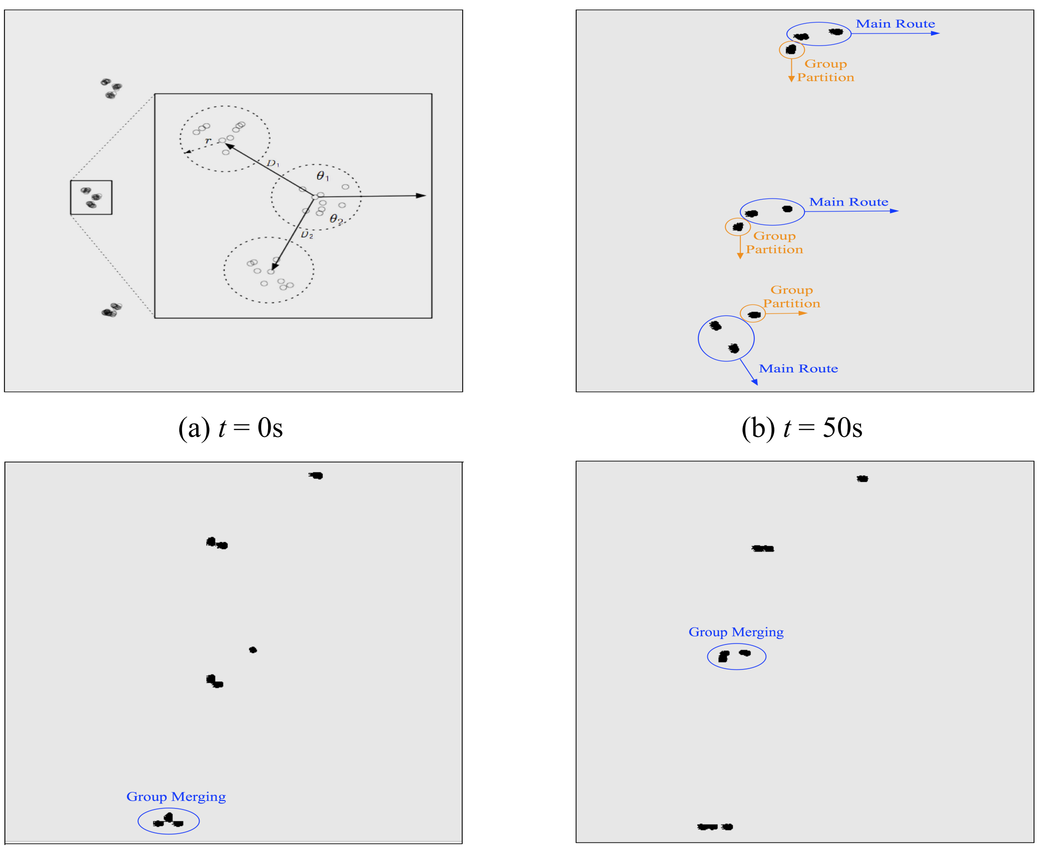
$$T_d = T_{d0} \times Blocking\ Levels, \qquad (4\text{-}2)$$

where T_{d0} is the initial delay time.

Both the blocking area and blocking level are randomly selected based on situations. If the situation is very serious or the enemy units offer stubborn resistance, the number of blocking areas are increased. If there is minor or little resistance, a small number of blocking areas exist.

4.5. Application of UMOMM

This subsection illustrates an example of a military urban operation under UMOMM. Figure 22 represents the group partitioning and merging mobility scenario. A company reaches the outside of an urban area and deploys its platoons as shown in Figure 22(a). After starting the simulation, a squad from each platoon gets partitioned and moves along a new route while the platoon leader and other squads keep their original routes, as in Figure 22(b). In Figure 22(c)~(d), the partitioned squads are merged with the main forces.



(a) t = 0s (b) t = 50s (c) t = 160s (d) t = 250s

Figure 22: Group partitioning and merging

5. Simulation Study and Results

The importance of the mobility model for evaluating the performance of routing protocols in MANETs has been discussed. In this section, the performance of routing protocols OLSR, DSR, and AODV using RWP, RPGM, and the proposed UMOMM are discussed. In order to accurately perform the simulation, several tools, including NS2, Network Animator (NAM), Tcl, and C++ programming were used.

5.1. Network Simulator (NS)

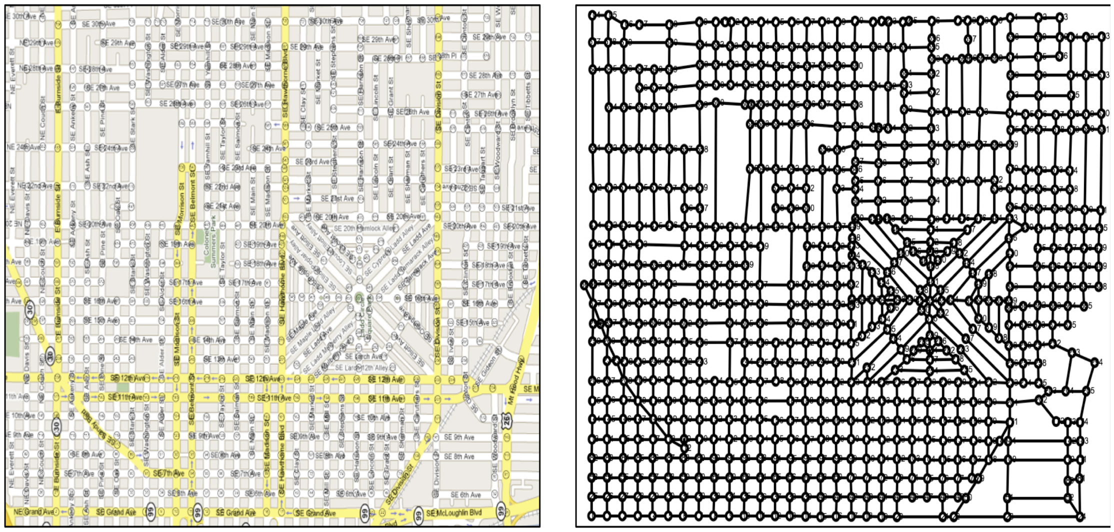
NS version 1 (NS1) is a discrete event simulator designed for networking research. It is open source software and allows for modification. The development of NS1 was supported by DARPA through the Virtual InterNetwork Testbed (VINT) project at LBL, Xerox PARC, UCB, and USC/ISI. In 1996, NS1 was extended to NS version 2 (NS2) with support from DARPA and NSF. NS2 also includes some code from the UCB Daedulus, CMU Monarch projects, and Sun Microsystems to support wireless network simulations.

NS adopts split-programming, which consists of C++ and OTcl (an object oriented extension of Tcl). C++ is used for route lookup, packet forwarding, and implementation of the TCP protocol to achieve time efficiency, whereas OTcl is used for aggregate statistics collection, modeling of link failures, route change, and low-rate control protocols to control the simulation [25].

To run NS2, a scenario file described by Tcl script is required. After running the simulation based on the scenario, NS2 generates several output files called trace files. These trace files include data on packet size, packet types, and packet events, such as packets sent, received, forwarded, and dropped. It also contains nodal movement logs for NAM.

5.2. Simulation environment

The simulation environment consists of a 2900×2300 meters urban area with an additional 50 meters buffer zone on all four sides for staging of the troops. The urban area is abstracted from a part of downtown Portland, Oregon, as shown in Figure 23(a). Figure 23(b) shows the path graph of Figure 23(a) containing 734 vertices. Our mobility model for the simulation studies have between 29 and 247 mobile nodes, depending on the scenarios.

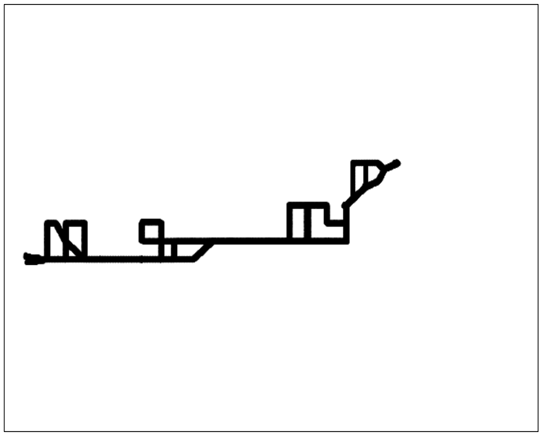


(a) A map of downtown Portland (b) Abstracted topology of Portland map

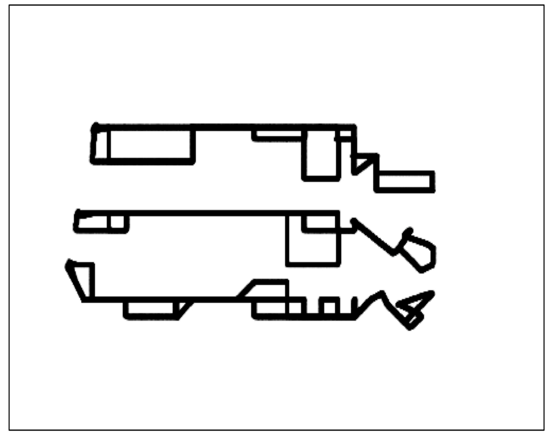
Figure 23: Simulation map and topology

There are four different scenarios, differing in the number of platoons; Scenario I has one platoon (27 soldiers), Scenario II has one company (3 platoons), Scenario III has two companies (6 platoons), and Scenario IV has three companies (9 platoons). In addition, each company has a commander and a battalion commander, who are deployed in the buffer zone to form a chain of command. Each platoon is associated with a mission, which involves searching for the enemy, gaining control of maneuver routes and intermediate locations, and occupying assigned locations on the map.

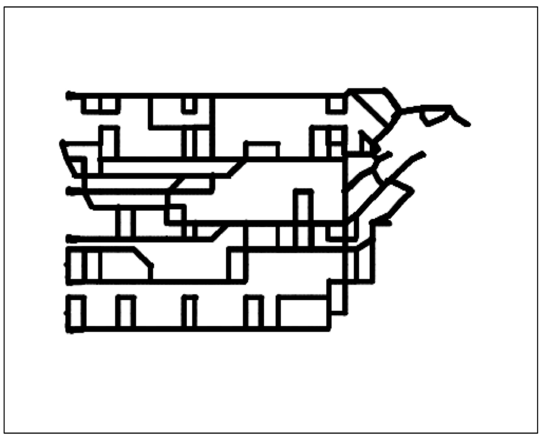
Figure 24 shows the different node movements for the four scenarios on the same map. Each platoon determines its route depending on its destination. The points where group partitioning and merging occur and the number of group partitioning are not specified beforehand. The platoon leader determines the partitioning and direction of movement at each intersection and also decides the merging point. Therefore, the movement patterns for the main route and the locations of group partitioning and merging are different for each scenario.



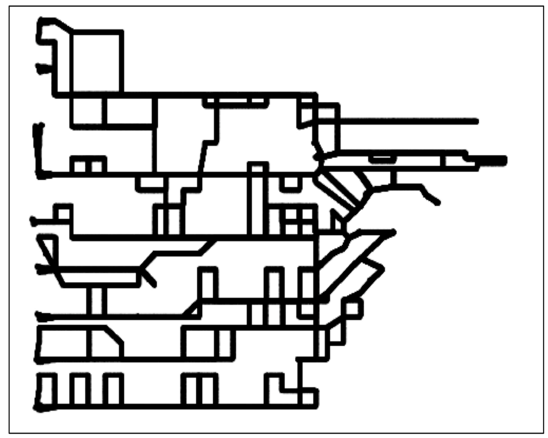
(a) One platoon (n=29)



(b) One company (3 platoons, n=83)



(c) Two companies (6 platoons, n=165)



(d) Three companies (9 platoons, n=247)

Figure 24: Trace of Node Movements

The traffic data are categorized into two types for military operations in an urban area: individual data and visual data. Individual data include information such as location, stress, fatigue, and vital signs of an individual soldier. On the other hand, visual data include live videos or pictures. During simulation, group members or individual soldiers send their information and visual data to their leader (squad or platoon leader). Then, a group leader will send data about the group members to its commander, and so on. Each group member is assumed to generate 8 Kbytes of data every 120 seconds, and the group leader generates 12 Kbytes of data every second. Table 1 summarizes the parameters for the simulation environment.

Table 1: Simulation environments

<table>
<tr><th colspan="2">Mobility Model</th><th>UMOMM</th><th>RPGM</th><th>RWP</th></tr>
<tr><td colspan="2">Simulation Area</td><td colspan="3">300m × 2400m</td></tr>
<tr><td colspan="2">Speed</td><td colspan="2">1.5~2.5m/s</td><td>0~2.5m/s</td></tr>
<tr><td rowspan="4">Scenario</td><td>I</td><td colspan="2">29 nodes
(27/platoon, 1 platoon)</td><td>29 nodes</td></tr>
<tr><td>II</td><td colspan="2">83 nodes
(27/platoon, 3 platoons)</td><td>83 nodes</td></tr>
<tr><td>III</td><td colspan="2">165 nodes
(27/platoon, 6 platoons)</td><td>165 nodes</td></tr>
<tr><td>IV</td><td colspan="2">247
(27/platoon, 9 platoons)</td><td>247</td></tr>
<tr><td colspan="2">Traffic Type</td><td colspan="3">CBR</td></tr>
<tr><td colspan="2">Simulation Time</td><td colspan="3">2000s</td></tr>
<tr><td colspan="2">Routing Protocols</td><td colspan="3">OLSR, DSR, AODV</td></tr>
</table>

As shown in Table 2, the blocking levels have been divided into the following 5 levels: barricades, light booby-trap or mine, heavy booby-trap or mine, light ambush, and heavy-ambush.

Table 2: Blocking level

Level	Blocking Area
1	Barricade
2	Light Booby-trap or Mine
3	Heavy Booby-trap or Mine
4	Light Ambush
5	Heavy Ambush

The performance of routing protocols using UMOMM is compared against RPGM and RWP. RPGM is chosen because it is a representative group mobility model that can be applied to military applications. On the other hand, RWP is the most widely used mobility model for wireless network simulation and evaluating routing protocols in MANETs. Unfortunately, RPGM does not support the group partitioning and merging and obstacle situations. Moreover, RWP does not model group activities, moving along streets, and overcoming obstacles.

RPGM was implemented using the mobility generation tool by F. Bai *et al.* [34], while RWP is provided by the *setdest* tool in the standard NS-2 distribution.

The mobility generation tool of RPGM has two parts: definition and movement of a group. For the definition of a group, input parameters that define the number of nodes in a group, and values for speed and angle deviation are required. For the movement of a group, each location and duration time needs to be defined. Figure 25 shows the parameters for the group leader's movements. The initial location of group leader is determined by <*ini_x*0> and <*ini_y*0>. The first destination is <*destination_x*1> and <*destination_y*1>, and <*duration_time_t*1> is the time to move from the initial location to the first destination. This process is repeated until the simulation ends. In order to generate group partitioning and merging and maintain squad movement, each group's route, with locations of intersections and duration times, needs to be specified individually.

<ini_x0> <ini_y0>

<destination_x1> <destination_y1> <duration_time_t1>

<destination_x2> <destination_y2> <duration_time_t2>

<destination_x3> <destination_y2> <duration_time_t3>

…

…

…

<destination_x(n)> <destination_y(n)> <duration_time_t(n)>

Figure 25: Parameters for RPGM mobility generator [34]

In order to implement the RWP mobility model, the *./setdest* command in NS-2 was used as shown in Figure 26, with the parameters in Table 1.

```
./setdest –v <1> -n <number of nodes> -p <pause time> -M <max speed>
-t <simulation time> -x <width of space> -y <height of space>
```

Figure 26: *Setdest* tool for RWP

5.3. Performance metrics

The performance evaluation of different routing protocols using different mobility models was performed based on the following three metrics: packet delivery ratio, average end-to-end delay, and normalized routing load. Figure 27 shows a simple program that gathers performance metrics using AWK.

5.3.1. Packet Delivery Ratio (PDR)

Packet delivery ratio (PDR) is defined as the ratio of the total number of packets received by all the destination nodes to the total number of packets sent by all the source nodes.

$$PDR = \frac{total\ number\ of\ received\ packets}{total\ number\ of\ sent\ packets} \tag{5-1}$$

PDR is an important measure of the efficiency of a routing protocol. A large PDR value indicates that destination nodes successfully received most of the packets that were sent and indicates high performance.

5.3.2. Average End-to-End Delay (AD)

Average end-to-end delay (AD) is the average time required for packets to travel from a source to a destination node. This does not include packets lost during transmission. AD is calculated as follow:

$$AD = \frac{\sum_{i=0}^{n}(time\ of\ received\ packet_i - time\ of\ sent\ packet_i)}{total\ number\ of\ received\ packets} \tag{5-2}$$

A large AD value indicates that the network is congested and the performance of packet transmission is low.

5.3.3. Normalized Routing Load

Normalized Routing Load (NRL) is the ratio of the total number of routing control packets sent to the total amount of data packets received by the nodes.

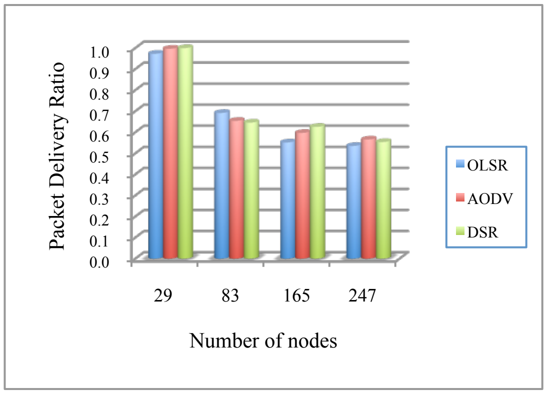
$$NRL = \frac{total\ number\ of\ sent\ routing\ packets}{total\ number\ of\ received\ data\ packets} \tag{5-3}$$

NRL is a measure of the routing overheads per data packet and also indicates the efficiency of a routing protocol. If NRL is high, the routing overhead increases for a routing protocol and hence the efficiency decreases.

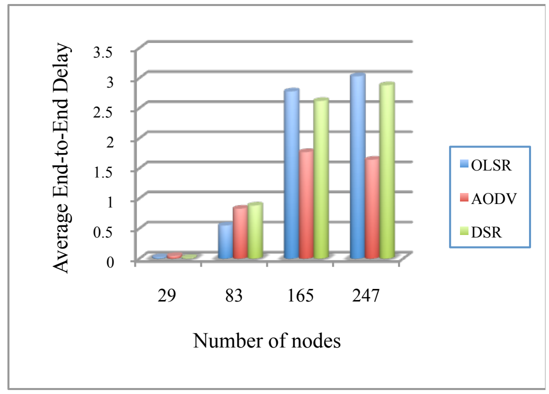
```
BEGIN { Max_Packet_Id = 0; Total_Time=0.0; Total_Receive=0;}
$1~/s/ && /AGT/ && /-It cbr/ {Data_Sent++}
$1~/s/ && /RTR/    {Routing_Sent++}
$1~/r/ && /AGT/ && /-It cbr/ {Data_Receive++}
{  action = $1;   time = $3;   Packet_ID = $41;
   if ( Packet_ID > Max_Packet_Id ) Max_Packet_Id = Packet_ID;
   if ( ($19 == "AGT") && (action == "s") ){
     start_time[Packet_ID] = time; }
   if ( ($1 == "r") && ($19 == "AGT")) {
                          Total_Receive += 1;
        end_time[Packet_ID] = time;
        Time=start_time[Packet_ID]-end_time[Packet_ID];
        Total_Time +=Time; } }
END {
            if (Data_Sent > 0) PDR = (Data_Receive/Data_Sent)*100;              # Packet Delivery Ratio
             AverDelay= Total_Time / Total_Receive;                             # Average end-to-end dealy
             if (Data_Receive > 0) NRL = (Routing_Sent/Data_Receive);    # Normalized Routing Load
    printf ("   Packet Delivery Ratio (Percentage):           %3.2f\n", PDR);
    printf ("   Average Dealy:                                %3.4f\n", AverDelay);
    printf ("   Normalized Routing Load (Percentage):      %3.5f\n", NRL);
}
```

Figure 27: A program of performance metrics

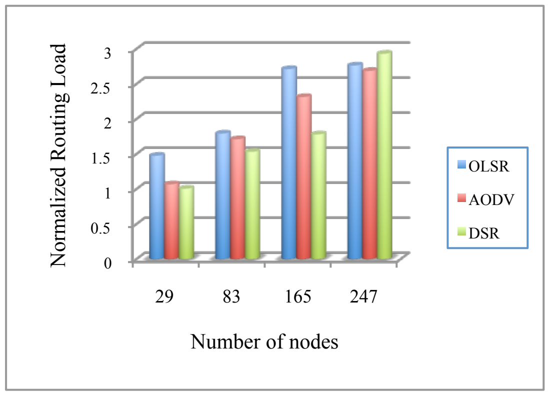
5.4. Simulation Results



(a) Packet Delivery Ratio



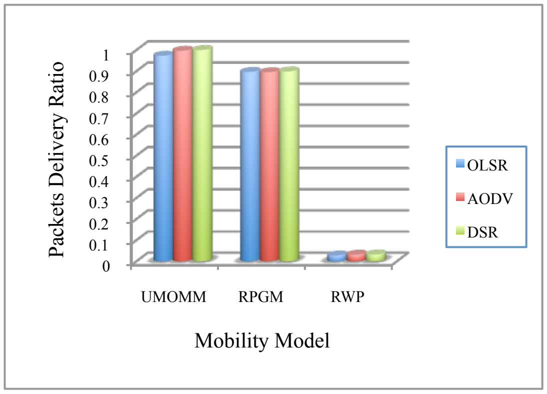
(b) Average End-to-End Delay



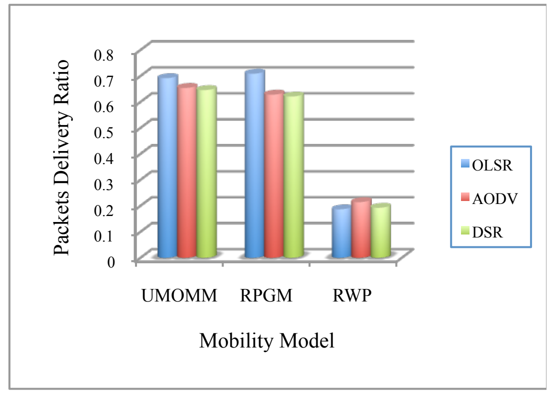
(c) Normalized Routing Load

Figure 28: Performance of UMOMM

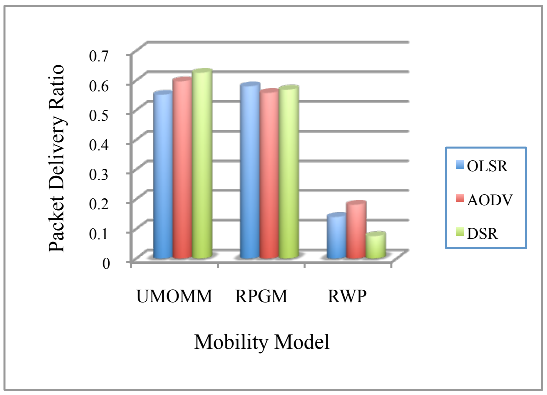
Figure 28 represents the performance of OLSR, DSR, and AODV routing protocols based on UMOMM. Figure 28(a) shows that the three routing protocols have a higher PDR when the number of nodes is small, and PDR decreases as the number of nodes increases. In contrast, as the number of nodes increases, the average end-to-end delay and the normalized routing load increase for both OLSR and DSR. For AODV, the average end-to-end delay and the normalized routing load gradually increase as a function of the number of nodes; however, AD decreases as the number of nodes increase from 165 to 247 nodes because only the packets that actually arrive at the destinations are considered. These simulation results show that the performance of AODV is pretty stable under varying node densities and is least sensitive to changes in group topology.



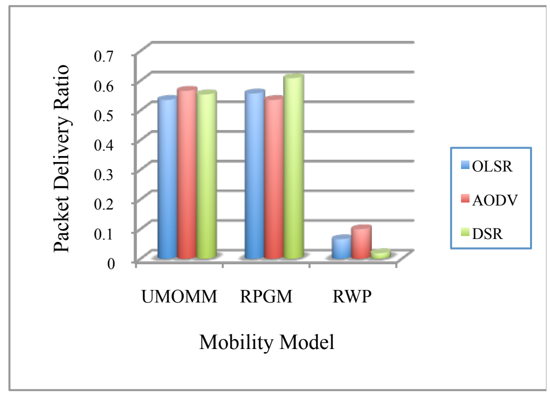
(a) $n = 29$



(b) $n = 83$



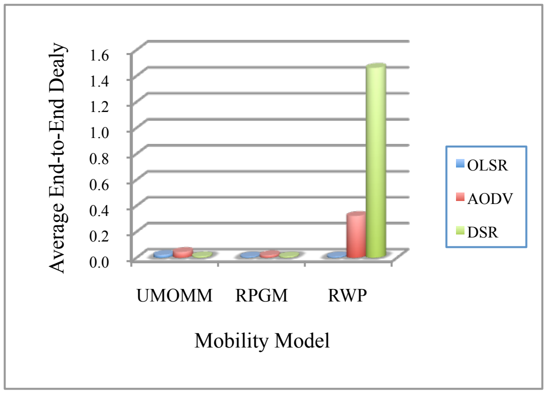
(c) $n = 165$



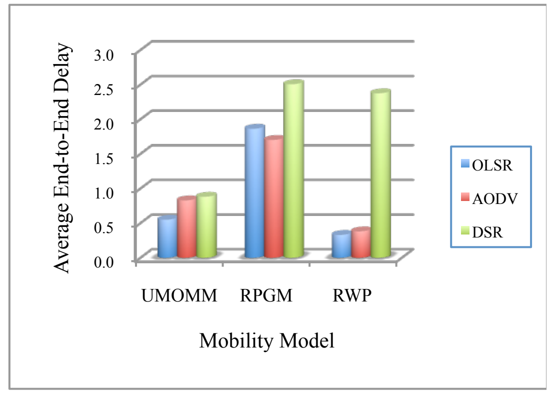
(d) $n = 247$

Figure 29: Packet Delivery Ratio

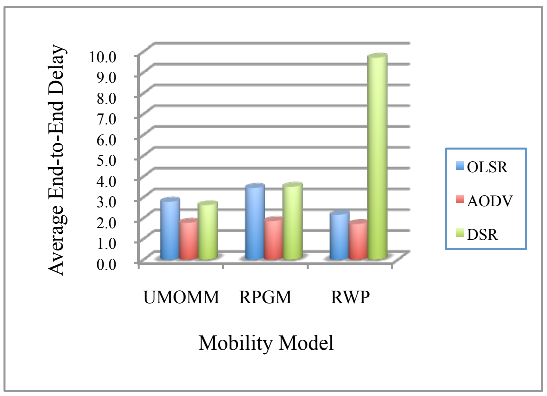
Figure 29 compares the performance of PDR for UMOMM, RPGM, and RWP. PDR for UMOMM and RPGM depend more on the mobility models than on routing protocols because the group leader determines the mobility of group members. The RWP model results in the worst performance because the mobility of nodes is not related to each other. Figure 29(a) shows that UMOMM provides the highest PDR when the number of nodes is 29. However, as the number of node increases the performance degrades because group partitioning and merging occurs frequently (see Figures 29(b)~(d)).



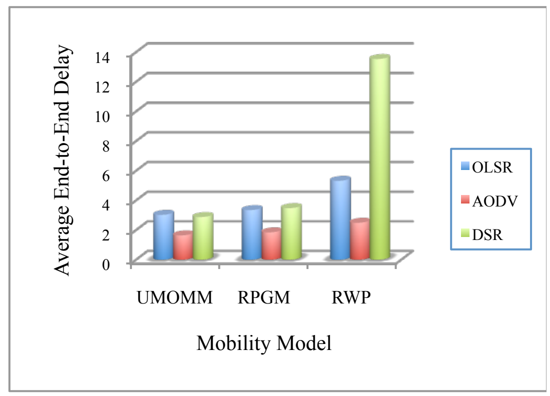
(a) *n* = 29



(b) *n* = 83



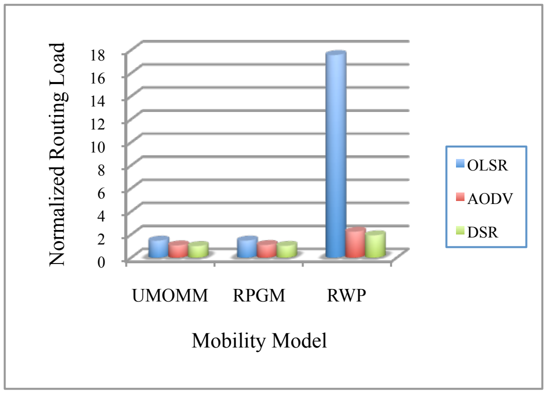
(c) n = 165



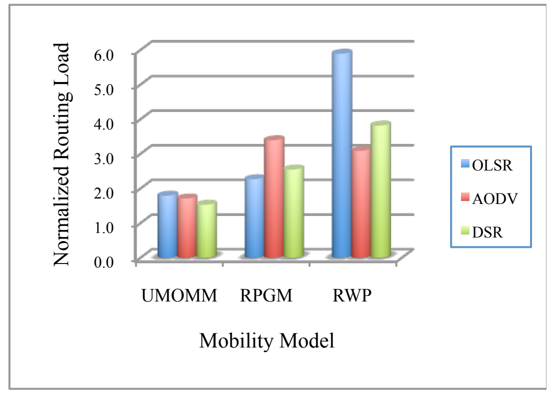
(d) n = 247

Figure 30: Average End-to-End Delay

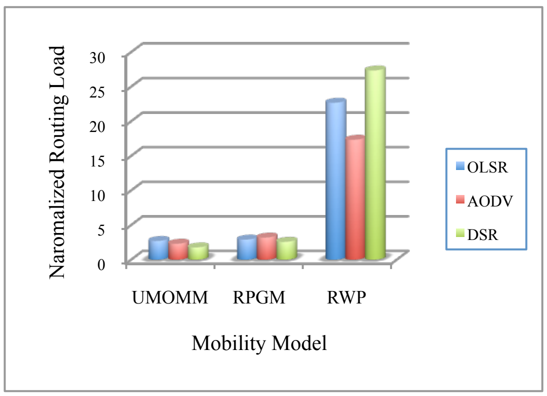
Figure 30 shows AD for the three routing protocols for varying number of nodes. AD increases as the number of nodes increases for all mobility models because the node density is increased. In particular, AD for RWP is significantly higher than for the others, which also leads to very low packet delivery ratio.



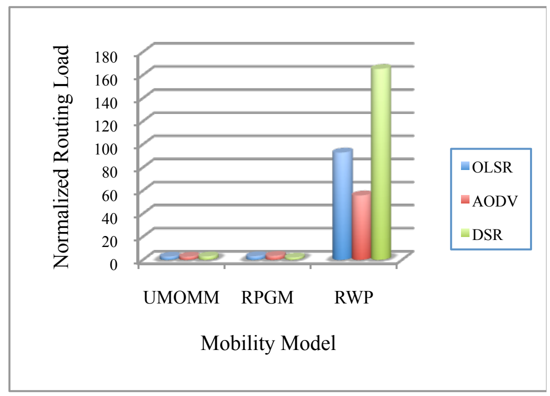
(a) n = 29



(b) n = 83



(c) n = 165



(d) n = 247

Figure 31: Normalized Routing Load

Figure 31 shows NRL as a function of number of nodes for all three routing protocols. The routing information for the entity model is not valid for very long because network topology changes quickly when many mobile nodes simultaneously move around the simulation area. Thus, NRL of RWP dramatically increases as the number of nodes increases. In contrast, the routing information for the group mobility model does not expire so quickly because nodes are distributed within certain distances. Therefore, both UMOMM and RPGM result in low NRL.

6. Conclusion and future work

As military units supported by MANETs perform missions in urban areas, a group may frequently partition when it is assigned new missions or approaches an intersection. A group leader must also determine the direction and merging point of a subgroup in order to allow for subgroups to regroup. As most group mobility models fail to describe such group mobility patterns, UMOMM was proposed to properly reflect movements in urban military operations. Our simulation results show that mobility models have an impact on the performance evaluation of routing protocols.

In this thesis, the network environment was assumed to be homogeneous, i.e., only the soldiers carry mobile devices. In order to develop a more realistic mobility model, a heterogeneous network must be considered that would include vehicles, aerial devices, and satellites.

BIBLIOGRAPHY

[1] X. Hong, M. Gerla, G. Pei, and C. Chiang, "A group mobility model for ad hoc wireless networks," *MSWiM '99: Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems,* Aug 1 1999.

[2] B. Zhou, K. Xu, and M. Gerla, "GROUP AND SWARM MOBILITY MODELS FOR AD HOC NETWORK SCENARIOS USING VIRTUAL TRACKS," *MILCOM,* Jan 1 2004.

[3] J. M. Ng and Y. Zhang;, "Reference region group mobility model for ad hoc networks," *Wireless and Optical Communications Networks, 2005. WOCN 2005. Second IFIP International Conference on,* pp. 290 - 294, Feb 28 2005.

[4] J. M. Ng and Y. Zhang, "A Mobility Model with Group Partitioning for Wireless Ad Hoc Networks," *Information Technology and Applications, 2005. ICITA 2005. Third International Conference on,* vol. 2, pp. 289 - 294, Jun 30 2005.

[5] J. M. Ng and Y. Zhang;, "Impact of group mobility on ad hoc networks routing protocols," *Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference,* vol. 2, pp. 6 pp. - 1134, Jan 31 2006.

[6] C. E. Perkins, "AD HOC NETWORKING," *ADDISON-WESLEY,* Jan 8 2001.

[7] D. S. Alberts, J. J. Garstka, and F. P. Stein, "Network Centric Warfare: Developing and Leveraging Infromation Superiority," *U.S. Department of Defense C4ISR Cooperative Research Program Publication Series,* 1999.

[8] J. B. Lyons, S. D. Swindler, and J. A. White, "Network centric warfare: Organizational collaboration as a key enabler," *Collaborative Technologies and Systems, 2008. CTS 2008. International Symposium on,* pp. 367 - 374, Apr 30 2008.

[9] E. M. Royer and C.-K. Toh;, "A review of current routing protocols for ad hoc mobile wireless networks," *Personal Communications, IEEE [see also IEEE Wireless Communications],* vol. 6, pp. 46 - 55, Apr 1 1999.

[10] M. Sanchez and P. Manzoni, "A Java based simulator for ad hoc networks," *Future generations computer systems,* vol. 17, p. 573~583, 2001.

[11] V. Vetriselvi and R. Parthasarathi, "Trace Based Mobility Model for Ad Hoc Networks," *Wireless and Mobile Computing, Networking and Communications, 2007. WiMOB 2007. Third IEEE International Conference on,* pp. 81 - 81, Sep 30 2007.

[12] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing,* Jan 1 2002.

[13] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile Computing, T. Imielinski and H. Korth, Eds. Norwell : Kluwer,* vol. 353, pp. 153-181, 1996.

[14] E. M. Royer, P. M. Melliar-Smith, and L. E. Moser, "An analysis of the optimum node density for ad hoc mobile networks," *Communications, 2001. ICC 2001. IEEE International Conference on,* vol. 3, pp. 857 - 861 vol.3, May 31 2001.

[15] K. H. Wang and B. Li;, "Group mobility and partition prediction in wireless ad-hoc networks," *Communications, 2002. ICC 2002. IEEE International Conference on,* vol. 2, pp. 1017 - 1021 vol.2, Jan 1 2002.

[16] L. Ning, G. Yan, D. Chao, and W. Jinlong;, "Diamond Group Mobility Model for ad hoc network in military," *Signal Processing, 2008. ICSP 2008. 9th International Conference on,* pp. 2754 - 2756, Sep 30 2008.

[17] K. Blakely and B. Lowekamp, "A structured group mobility model for the simulation of mobile ad hoc networks," *Proceedings of the second international workshop on Mobility management & wireless access protocols,* Jan 1 2004.

[18] C. E. Perkins and P. Bhagwat, "Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers," *ACM SIGCOMM Computer Communication Review,* vol. 24, Oct 1 1994.

[19] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol(OLSR)." *RFC 3626, IETF Network Working Group,* Oct. 2003.

[20] D. Medhi and K. Ramasamy, "Network Routing: Algorithms, Protocols, and Architectures," *Morgan Kaufmann,* Mar 29 2007.

[21] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," *Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International,* pp. 62 - 68, Nov 30 2001.

[22] Z. Huawei and Z. Yun;, "Comparison and Analysis AODV and OLSR Routing Protocols in Ad Hoc Network," *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on,* pp. 1 - 4, Sep 30 2008.

[23] D. Johnson, Y. Hu, and D. Maltz, "The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4," *draft-ietf-manet-dsr-10.txt,* Feb 2007.

[24] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," *draft-ietf-manet-aodv-13.txt,* Feb 17 2003.

[25] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, and H. Yu;, "Advances in network simulation," *Computer,* vol. 33, pp. 59 - 67, May 1 2000.

[26] C. Bettstetter, "Mobility modeling in wireless networks: categorization, smooth movement, and border effects," *ACM SIGMOBILE Mobile Computing and Communications Review,* Jan 1 2001.

[27] C. Bettstetter and C. Wagner, "The spatial node distribution of the random waypoint mobility model," *German Workshop on Mobile Ad Hoc Networks (WMAN),* Jan 1 2002.

[28] F. Bai, N. Sadagopan, and A. Helmy, "IMPORTANT: A framework for analyzing the Impact of Mobility on Performance Of RouTing protocols for Adhoc NeTworks," *Ad Hoc Networks,* Jan 1 2003.

[29] J. Tian, L. Han, K. Rothermel, and C. Cseh, "Spatially aware packet routing for mobile ad hoc inter-vehicle radio networks," *Proc. of the IEEE Intelligent Transportation System (ITSC),* Jan 1 2003.

[30] J. Tian, J. Hahner, C. Becker, I. Stepanov, and K. Rothermel, "Graph-based mobility model for mobile ad hoc network simulation," *Simulation Symposium, 2002. Proceedings. 35th Annual,* pp. 337 - 344, Mar 31 2002.

[31] E. Gerhards-Padilla, M. Gerharz, M. Frank, and P. Martini, "Modelling mobility in disaster area scenarios," *Proceedings of the 10th ACM Symposium on Modeling,* Jan 1 2007.

[32] B. Divecha, A. Abraham, C. Grosan, and S. Sanyal, "Analysis of Dynamic Source Routing and Destination-Sequenced Distance-Vector Protocols for Different Mobility Models," *Modelling & Simulation, 2007. AMS '07. First Asia International Conference on,* pp. 224 - 229, Feb 28 2007.

[33] S. Reidt and S. D. Wolthusen, "An Evaluation of Cluster Head TA Distribution Mechanisms in Tactical MANET Environments," *Proc. of International Technical Alliance in Network and ...,* Jan 1 2007.

[34] F. Bai, N. Sadagopan, and A. Helmy, "IMPORTANT Mobility Tool Generators in ns-2 simulator," [Online]. Available : http://nile.usc.edu/important/ Feb. 1 2004.