A QUEUING MECHANISM FOR H.264 VIDEO IN MANET

A Thesis
Presented to the
Faculty of
San Diego State University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Electrical Engineering

by
Manoj Prabhakar Goud Gandamalla
Spring 2013
SAN DIEGO STATE UNIVERSITY

The Undersigned Faculty Committee Approves the
Thesis of Manoj Prabhakar Goud Gandamalla:

A Queuing Mechanism for H.264 Video in Manet

Sunil Kumar, Chair
Department of Electrical and Computer Engineering

Satish K. Sharma,
Department of Electrical and Computer Engineering

Usha Sinha
Department of Physics

Dec 12, 2012
Approval Date
DEDICATION

My family
ABSTRACT OF THE THESIS

A Queuing Mechanism for H.264 Video in MANET
by
Manoj Prabhakar Goud Gandamalla
Master of Science in Electrical Engineering
San Diego State University, 2013

Mobile ad-hoc network (MANET) is a collection of mobile nodes, which form a wireless network without the use of an existing infrastructure. MANET’s are usually characterized by mobility of nodes and communication over multiple hops. Hence, MANET links are short lived and the topology is very dynamic.

In order to simulate a new protocol for MANET, it is necessary to use a mobility model that accurately represents the mobile nodes (MN’s) that will eventually utilize the given protocol. Only in this type of scenario is it possible to determine whether or not the proposed protocol will be useful when implemented. The existing mobility model generator are random and do not depict the scenarios that are needed for successful evaluation of a protocol for MANET. Hence we propose a new mobility model generator for MANET mobility and discuss few proposed scenarios generated for emergency scenarios.

In later part we discuss the challenges of streaming multimedia over MANET’s. Extensive research has already been done to ensure a smooth and uninterrupted multimedia transmission to a mobile node (MN). Before sending packets into the packets, video is compressed by the encoder by reducing/removing redundancies from the video data. After compression different video packets have relatively different importance. MANETs do not have the ability to serve the multimedia information based on this relative importance of the packets. Hence, we propose a Queuing mechanism to consider the relative importance of this multimedia information and achieve a Quality of Service (QoS) for transmission of Multimedia information over Mobile ad-hoc network.

The simulation results obtained have shown that the higher priority packets have achieved better throughput compared to relatively lower priority packets. This will result in a better streaming quality of multimedia information over MANET’s.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>v</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>ix</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>x</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>xii</td>
</tr>
<tr>
<td>ACRONYMS</td>
<td>xiii</td>
</tr>
<tr>
<td>1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Motivation of Thesis</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Research Contributions</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Thesis Outline</td>
<td>3</td>
</tr>
<tr>
<td>2 INTRODUCTION TO ROUTING PROTOCOLS AND H.264 VIDEO CODING STANDARD</td>
<td>4</td>
</tr>
<tr>
<td>2.1 Introduction to AODV</td>
<td>4</td>
</tr>
<tr>
<td>2.2 Introduction to H.264 Video Coding Standard</td>
<td>5</td>
</tr>
<tr>
<td>2.2.1 VCL</td>
<td>6</td>
</tr>
<tr>
<td>2.2.2 Slices and Slice Groups</td>
<td>6</td>
</tr>
<tr>
<td>2.3 Flexible Macroblock Ordering (FMO)</td>
<td>7</td>
</tr>
<tr>
<td>3 MOBILITY MODELS</td>
<td>9</td>
</tr>
<tr>
<td>3.1 Classification of Mobility Models</td>
<td>9</td>
</tr>
<tr>
<td>3.2 Random Walk Mobility Model</td>
<td>10</td>
</tr>
<tr>
<td>3.3 Random Waypoint Mobility Model (RWPM)</td>
<td>11</td>
</tr>
<tr>
<td>3.3.1 Common Problem with RWPM</td>
<td>12</td>
</tr>
<tr>
<td>3.3.2 RWPM with Attraction Points</td>
<td>12</td>
</tr>
<tr>
<td>3.4 Random Direction Mobility Model</td>
<td>12</td>
</tr>
<tr>
<td>3.5 Gauss-Markov Mobility Model</td>
<td>14</td>
</tr>
<tr>
<td>3.6 A Probabilistic Version of Random Walk</td>
<td>14</td>
</tr>
<tr>
<td>3.7 City Section Mobility Model</td>
<td>16</td>
</tr>
</tbody>
</table>
3.8 Group Mobility Models ................................................................. 17
  3.8.1 Column Mobility Model ............................................................. 17
  3.8.2 Nomadic Community Mobility Model ................................. 18
  3.8.3 Pursue Mobility Model .............................................................. 18
  3.8.4 Reference Point Group Mobility Model .................................. 19
3.9 Mobility Model for Emergency Scenarios ....................................... 20
  3.9.1 Virtual Track Based Group Mobility Model ....................... 21
  3.9.2 A Flocking-Based Mobility Model .......................................... 21
3.10 Mobility Models for UAV Applications ......................................... 23
  3.10.1 UAV Movement ..................................................................... 23
  3.10.2 Hover above Movement UAV Protocol ............................ 23
  3.10.3 Landmark Routing in Large Wireless Battlefield Networks
        Using UAVs ............................................................................... 24
  3.10.4 Semi-Random Circular Movement Model ............................ 25
  3.10.5 Mobility Models for UAV Group Reconnaissance Applications .. 26
3.11 Motivation for a New Mobility Model Generator .......................... 27
  3.11.1 Emergency Area Mobility Model ........................................... 28
  3.11.2 Multiple Emergency Areas with Group Mobility ................ 28
  3.11.3 Mobility Model for UAV Surveillance ................................. 30
  3.11.4 Circular Motion for UAV Surveillance and Emergency
        Operations .................................................................................. 30
  3.11.5 Elliptical Mobility Model for UAV Surveillance .................... 31
4 QUEUING MECHANISM ........................................................................ 33
  4.1 Related Work .................................................................................. 33
  4.2 H264/AVC Video Packet Prioritization ....................................... 36
  4.3 Queuing Schemes in NS2 ................................................................. 36
    4.3.1 InterFace Queue ...................................................................... 37
    4.3.2 Event Scheduler .................................................................... 38
  4.3 Priority Issue .................................................................................. 38
  4.4 Time Stamp Issue ......................................................................... 40
  4.5 Queue Overflow ............................................................................ 40
  4.6 Lockout .......................................................................................... 42
  4.7 Proposed Queuing Scheme for H.264 Video ............................... 42
4.7.1 At Enque ..................................................................................................................42
  4.7.1.1 Check for Time to Live (TTL) while En-Queuing the Packet ........................................42
  4.7.1.2 Check for Buffer Fullness .....................................................................................43
4.7.2 Queue Sorting ...........................................................................................................44
  4.7.2.1 Frame Number Based Sorting .............................................................................44
  4.7.2.2 Priority Based Sorting .........................................................................................45
4.8 At Dequeue ..................................................................................................................46
  4.8.1 Data Rate Control ...................................................................................................47
  4.8.2 Check for TTL before Dequeing a Packet ..............................................................48
4.9 Results ..........................................................................................................................48

5 CONCLUSION ................................................................................................................53
REFERENCES ....................................................................................................................54
APPENDIX
  MOBILITY MODEL GENERATOR .......................................................................................57
LIST OF TABLES

Table 3.1. UAV Random Action Table ...................................................................................26
Table 3.2. UAV Pheromone Action Table ..............................................................................27
Table 4.1. Network Scenario ...................................................................................................48
LIST OF FIGURES

Figure 2.1. Structure of H.264/AVC Encoder. .................................................................6
Figure 2.2. Division of picture into slices without the use of FMO. .................................7
Figure 3.1. Travel pattern of an MN using the 2-D Random Walk Mobility Model. ........11
Figure 3.2. Traveling pattern of a node using the Random Waypoint Mobility Model. ....12
Figure 3.3. Traveling pattern of an MN using the Random Direction Mobility Model. ....13
Figure 3.4. Travelling pattern of a node using the Gauss-Markov Mobility Model.........14
Figure 3.5. Flow chart of the probabilistic version of random walk..............................15
Figure 3.6. Example of traveling pattern of an MN using the probabilistic version of the Random Walk Mobility Model.........................................................15
Figure 3.7. Traveling pattern of an MN using a probabilistic version of Random Walk. ..................................................................................................................16
Figure 3.8. Movements of three MNs using the Column Mobility Model. .....................16
Figure 3.9. Movements of seven MNs using the Nomadic Community Mobility Model. .........................................................................................................................18
Figure 3.10. Movements of six MNs using the Pursue Mobility Model. .........................19
Figure 3.11. Movements of three MNs using the RPGM model. .....................................20
Figure 3.12. Traveling pattern of five groups using the RPGM model. .........................20
Figure 3.13. Overview of Virtual Track Based Group Mobility Model. .........................21
Figure 3.14. The neighborhood of a unit is defined by the angle and the distance r. .......22
Figure 3.15. Flocking rules. ..............................................................................................22
Figure 3.16. Simulation snapshot of Platoon on the ground (grey and black nodes) and 3 drones in 50 meters height (big orange nodes). .............................................24
Figure 3.17. LANMAR routing in a “flat” MANET without UAVs. .................................24
Figure 3.18. Multi-level hierarchical ad hoc network.......................................................25
Figure 3.19. LANMAR routing in UAV based hierarchical ad hoc networks. ................25
Figure 3.20. Pheromone search pattern...........................................................................27
Figure 3.21. Emergency area mobility model.................................................................29
Figure 3.22. Multiple emergency areas with group mobility. (a) Nodes form 10 groups of 5 nodes each and divide into four and six groups to start towards
destination 1 and 2 respectively. (b) Groups reach their respective
destinations. (c) When a new emergency area (3) emerges two groups stick to
(2) whereas four groups move towards (3). .................................................................29

Figure 3.23. UAV surveillance mobility model (a) UAVs start toward their assigned
emergency areas. (b) UAVs steering towards their assigned emergency area’s
(c) UAVs hovering over the assigned emergency areas .............................................31

Figure 3.24. (a)(b)(c)(d) Collectively shows circular motion of Node ‘0’. .........................31

Figure 3.25. (a)(b)(c) Depict the elliptical mobility of nodes (0-5) in clockwise
direction and nodes (6-11) in anti-clockwise direction ...........................................32

Figure 4.1. Queuing scheme in NS2 .......................................................................................37
Figure 4.2. Flow chart of NS2 queuing logic ........................................................................38
Figure 4.3. Event scheduler ..................................................................................................39
Figure 4.4. Importance of priority .......................................................................................40
Figure 4.5. Importance of timestamp ..................................................................................41
Figure 4.6. Queue overflow ..................................................................................................41
Figure 4.7. Queue condition while en-queuing a packet .......................................................43
Figure 4.8. Enqueing packets in a sorted fashion .................................................................45
Figure 4.9. Dequeuing a packet ...........................................................................................47

Figure 4.10. Packet delivery ratio using different priorities as a function of varying
number of connections at average node speed of at $V = 5\text{m/s}$. Note that ns-2
uses a default drop-tail queuing mechanism which is priority-agnostic ......................49

Figure 4.11. Packet delivery ratio using different priorities as a function of varying
number of connections at average node speed of at $V = 10\text{m/s}$. Note that ns-2
uses a default drop-tail queuing mechanism which is priority-agnostic ....................50

Figure 4.12. Packet delivery ratio using different priorities as a function of varying
number of connections at average node speed of at $V = 10\text{m/s}$. Note that ns-2
uses a default drop-tail queuing mechanism which is priority-agnostic ......................51
ACKNOWLEDGEMENTS

I sincerely thank my advisor, Prof. Sunil Kumar for his valuable guidance and encouragement throughout the time I have been working on my thesis. He helped me to have confidence and be patient when things appeared to go south.

I would also like to thank my committee members Dr. Satish Sharma and Dr. Usha Sinha for agreeing to be a part of my thesis committee and providing valuable suggestions towards improvement of my thesis.

A special thanks to my lab mates Kashyap Kambatla, Sandeep Nanda and Kalyani Girvanesh for sharing their valuable knowledge and being a part of many intellectual discussions which helped me streamline my thesis progress. I also want to thank my other lab mates who helped me rejuvenate my energy with general gossip during my coffee breaks.

Finally I would like to thank my family AMMA, NANNA, and my dear FRIENDS for giving me patience and strength for the entire episode.
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MANET</td>
<td>Mobile ad-hoc network</td>
</tr>
<tr>
<td>MN</td>
<td>Mobile Node</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>UAV</td>
<td>Unmanned aerial vehicle</td>
</tr>
<tr>
<td>UGV</td>
<td>Unmanned ground vehicle</td>
</tr>
<tr>
<td>LANMAR</td>
<td>Landmark Routing</td>
</tr>
<tr>
<td>AODV</td>
<td>Ad-hoc on demand distance vector routing</td>
</tr>
<tr>
<td>RREQ</td>
<td>Route request</td>
</tr>
<tr>
<td>RREP</td>
<td>Route reply</td>
</tr>
<tr>
<td>JVT</td>
<td>Joint video team</td>
</tr>
<tr>
<td>FMO</td>
<td>Flexible macroblock ordering</td>
</tr>
<tr>
<td>RWMM</td>
<td>Random Walk Mobility Model</td>
</tr>
<tr>
<td>RWPM</td>
<td>Random Waypoint Mobility Model</td>
</tr>
<tr>
<td>RDMM</td>
<td>Random Direction Mobility Model</td>
</tr>
<tr>
<td>GMMM</td>
<td>Gauss-Markov Mobility Model</td>
</tr>
<tr>
<td>CSMM</td>
<td>City Section Mobility Model</td>
</tr>
<tr>
<td>CMM</td>
<td>Column Mobility Model</td>
</tr>
<tr>
<td>NCMM</td>
<td>Nomadic Community Mobility Model</td>
</tr>
<tr>
<td>PMM</td>
<td>Pursue Mobility Model</td>
</tr>
<tr>
<td>RM</td>
<td>Random motion vector</td>
</tr>
<tr>
<td>SRCMM</td>
<td>Semi-Random Circular Movement Model</td>
</tr>
<tr>
<td>GOP</td>
<td>Group of pictures</td>
</tr>
<tr>
<td>PLR</td>
<td>Packet loss rate</td>
</tr>
<tr>
<td>TTL</td>
<td>Time to live</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Full Form</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>IFQ</td>
<td>Interface Queue</td>
</tr>
<tr>
<td>MAC</td>
<td>Medium access control</td>
</tr>
<tr>
<td>FIFO</td>
<td>First in first out</td>
</tr>
<tr>
<td>FQ</td>
<td>Fair queueing</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

MANET is a collection of mobile nodes, which form a wireless network without the use of existing infrastructure. MANET’s are usually characterized by mobility of nodes and communication over multiple hops. These characteristics make MANET links to be intermittent and the topology to be dynamic. The wireless devices or mobile nodes in MANET are capable of self-configuring and communicating with each other without the need of any existing backbone infrastructure.

The increasing use of mobile devices and the demand for mobile multimedia applications are becoming increasingly popular in recent years due to explosive growth in their demand in both wired and wireless networks. Many advances in the video compression technology have been made in recent years resulting in the development of JVT H.264/MPEG-4 Part 10 that offers an enhanced video technology. This standard provides superior compression performance and better error resiliency, as well as many other features. H.264/AVC is a widely used standard for video streaming because of its high compression efficiency, robustness against errors and network-friendly features. However, providing the desired quality of service (QoS) or improving the transmission efficiency for H.264 video transmissions over wireless networks present many challenges. In the following sections we discuss the challenges and a queuing scheme to improve the QoS of the video transmission over MANET.

1.1 MOTIVATION OF THESIS

One of the major features of MANETs is mobility along with variable transmission range, variable processing capability and their ability to join and leave the network arbitrarily [1]. In such a scenario the job of a routing scheme is to successfully and efficiently deliver data packets from the source to the destination, which is further complicated with absence of known topology and the lack of a central controlling node.
In such situations to evaluate a routing protocol or to come up with a scheme which will be used for analyzing the behavior of MANETs, a widely used Random way-point (RWP) mobility model is not sufficient. Hence, we have designed a mobility model generator and propose few mobility patterns which include ground troops and UAVs for emergency scenarios and reconnaissance operations.

H.264 video codec is very sensitive to impairments due to transmission over noisy channels because the compressed data has very high entropy. Most error resiliency/concealment techniques provided in the H.264 standard deal with packet losses. Even with all these built-in techniques, the end-user experience may deteriorate in hostile wireless environments. The main idea behind H.264 is the use of coded pictures which can be ranked according to their importance for the quality of the received video sequence. For example, non-reference pictures, such as conventional B pictures, are least important, as their absence does not affect decoding of any other pictures. Inherently MANETs do not have the ability to serve the multimedia information based on the relative importance of these frames. Moreover, it also has to be able to satisfy the QoS requirements of real-time applications such as connections with delay-sensitivity and low packet loss rate [2].

Since the packet drops in MANET are inevitable, it is a good idea to use the available bandwidth in an efficient manner by prioritizing the relatively more important packets. Hence, we propose a Queuing mechanism to consider the relative importance of this multimedia information and achieve a QoS for transmission of multimedia information over MANETs.

1.2 RESEARCH CONTRIBUTIONS

The major contributions of this thesis are as follows:

1. Developed a new ‘mobility model generator’ which is compatible with NS2. This model can be useful for many applications including the ground troop movements, vehicles, and unmanned aerial vehicles (UAVs).

2. Proposed new mobility patterns for different scenarios. These models represent the patterns that are followed by the troops and the UAVs during emergency and surveillance scenarios.

3. Developed a smart queuing mechanism to support the QoS demands of video transmission over MANETs. For H.264 video transmission, the packet scheduling is done smartly so that the percentage of successfully transmitted more important packets (such as instantaneous decode refresh (IDR) frames/slices) increases.
designed a “queuing strategy” to give higher channel access to relatively more important packets, considering their frame number, priority, and time-to-live (TTL) values as the important constraints. This significantly improved the throughput of the important H.264 video slices and packets.

1.3 Thesis Outline

This thesis contains 5 chapters as discussed below:

- Chapter 1 has a brief introduction and lists the challenges of multimedia transmission over MANETs.
- Chapter 2 introduces the AODV routing protocol and reviews the H.264 video coding standard. These are used in the later sections to analyze the proposed QoS scheme.
- Chapter 3 presents a detailed analysis of existing mobility models and proposes a new mobility model generator for emergency and aerial surveillance scenarios.
- Chapter 4 proposes our smart queuing mechanism to improve the throughput of important packets compared to the relatively less important packets.
- Chapter 5 discusses conclusions, summarizes the research effort, and proposes the future work.
CHAPTER 2

INTRODUCTION TO ROUTING PROTOCOLS
AND H.264 VIDEO CODING STANDARD

Routing schemes are a part of the network layer for deciding on which output channel/route an incoming packet should be transmitted. Since MANET links are intermittent due to the dynamic topology, many on-demand and robust routing protocols have been developed. These are broadly classified as pro-active and re-active routing protocols. Pro-active protocols, as the name suggests, always have a route to any node within the network, which comes with an added overhead to maintain the routes. On the other hand, the reactive protocols form routes on-demand basis. Since routes for MANETs are intermittent, reactive protocols are the ideal choice. Since this thesis work uses the ad-hoc on demand distance vector routing (AODV) protocol to evaluate the proposed QoS scheme for MANETs, we briefly discuss below this routing protocol.

2.1 INTRODUCTION TO AODV

AODV, which is one of the most discussed reactive routing protocols, uses a broadcast route discovery mechanism [3]. The protocol relies on dynamically establishing route table entries at intermediate nodes. To maintain the most recent routing information between nodes, it borrows the concept of destination sequence numbers from DSDV [4]. The protocol also uses the route discovery mechanism which was also used in the Dynamic Source Routing (DSR) scheme. The combination of these techniques yields an algorithm that uses bandwidth efficiently by minimizing the network load for control and data traffic, is responsive to changes in topology, and ensures loop-free routing.

Path discovery also termed as route discovery is initiated whenever a source node needs to communicate with another node for which it has no routing information in its table. The source node initiates the route discovery process by broadcasting the route request packet, also termed as RREQ, to its neighbors.

The RREQ packet consists of:

1. Source_addr, i.e., Source IP address
2. Source_sequence_#, i.e., Source sequence number
3. Broadcast_id, i.e., unique broadcast ID
4. Dest_addr, i.e., Destination IP address
5. Dest_sequence_#, i.e., Destination sequence number
6. Hop_cnt, i.e., Hop count

The sequence number and the broadcast id are maintained as two separate counters. The pair (source_addr, broadcast_id) uniquely identifies a RREQ, and the broadcast_id is incremented whenever the source issues a new RREQ. When the intermediate node receiving the RREQ packet checks for the address in the packets, if the particular node is the intended destination it sends an RREP (Route Reply) else it just re-broadcasts the RREQ packet to the next intermediate node in the neighborhood after increasing the hop_cnt (hop count). Being in the connection of many nodes in the neighborhood, there is the possibility that the particular intermediate node may receive the same RREQ again. In order to avoid multiple duplicate packet acceptances the node checks for the broadcast_id and source address, and drops the packet if it has already received or forwarded the packet.

If a particular node does not satisfy the RREQ requirement, i.e., if it’s not the intended destination, the node keeps track of the forward path of the packet in order to set up reverse path to the source from the destination in case of transmission of RREP. This concept has an advantage of setting up two way transmission path; forward path and reverse path.

### 2.2 INTRODUCTION TO H.264 VIDEO CODING STANDARD

H.264/AVC is a state-of-the-art video coding standard developed by the Joint Video Team (JVT). Its enhanced compression performance and “network friendliness” make this standard very popular [5].

Conceptually, H.264 encoder is divided into Video Coding Layer (VCL) and Network Abstraction Layer (NAL). VCL is designed to efficiently represent the video content and NAL to format the VCL representation so as to be compatible with various transport streams or storage media.

Figure 2.1 illustrates the structure of an H.264/AVC video encoder. VCL generates the coded macroblocks (MB). These MBs are aggregated to form slices at the NAL by exploiting context adaptive coding. The H.264 slices consist of macroblocks processed in raster scan order when not using FMO. Slices formed at the NAL can be classified into 5
different priorities based on their properties. Each slice can be considered as a packet ready
to be transmitted over a network [5].

2.2.1 VCL

The VCL design follows the so-called block based hybrid video coding approach, in
which each coded picture is represented in block-shaped units of associated luma and chroma
samples called macroblocks. The basic source-coding algorithm is a hybrid of inter-picture
prediction to exploit temporal statistical dependencies and transform coding of the prediction
residual to exploit spatial statistical dependencies. There is no single coding element in the
VCL that provides the majority of the significant improvement in compression efficiency in
relation to prior video coding standards. It is rather a plurality of smaller improvements that
add up to the significant gain [5].

2.2.2 Slices and Slice Groups

Slices are a sequence of MBs which are processed in the order of a raster scan when
not using FMO. A picture may be split into one or several slices as shown in Figure 2.2 [5].
A picture is therefore a collection of one or more slices in H.264/AVC. Slices are self-
contained in the sense that given the active sequence and picture parameter sets, their syntax
elements can be parsed from the bit stream and the values of the samples in the area of the
picture that the slice represents can be correctly decoded without use of data from other slices
provided that utilized reference pictures are identical at encoder and decoder. Some
information from other slices may be needed to apply the de-blocking filter across slice
boundaries.
Regardless of whether FMO is in use or not, each slice can be coded using different coding types as follows.

- **I slice**: A slice in which all macroblocks of the slice are coded using intra prediction.
- **P slice**: In addition to the coding types of I slice, some macroblocks of the P slice can also be coded using inter prediction with at most one motion-compensated prediction signal per prediction block.
- **B slice**: In addition to the coding types available in a P slice, some macroblocks of the B slice can also be coded using inter prediction with two motion-compensated prediction signals per prediction block.

### 2.3 Flexible Macroblock Ordering (FMO)

H.264 provides the possibility of increasing the resilience of a coded stream by means of FMO. Using FMO, each macroblock can be assigned freely to a specific slice group using a macroblock allocation map (MBA_map). FMO provides great flexibility in terms of defining the coding order of macroblocks within a picture. H.264/AVC allows defining of a maximum of eight slice groups in one picture and the macroblocks within a certain slice group can further be subdivided into a number of slices. The case where there is only one slice group within a picture is similar to case of using no FMO. The power of FMO as an error resiliency tool lies within the way the macroblocks are ordered. For instance, slice groups can be constructed independently from each other and a picture can be decoded even if not all the slice groups are available or even if some slices are missing from one or more slice groups since it is possible to use information of the surrounding macroblocks.
important quality of FMO-based error resiliency is that it is particularly well suited to real-
time and low-delay applications like video conferencing.
CHAPTER 3

MOBILITY MODELS

In order to simulate a new protocol for MANET, it is imperative to use a mobility model that accurately represents the MN’s that will eventually utilize the given protocol. Only in this type of scenario is it possible to determine whether or not the proposed protocol will be useful when implemented.

3.1 CLASSIFICATION OF MOBILITY MODELS

Following are the types of mobility models used in the simulation of networks [6]:

Traces are the mobility patterns that are observed in real life systems. Traces provide accurate information, especially when they involve a large number of participants and over a long observation period. However, new network environments (e.g., MANET) are not easily modeled if traces have not yet been created. In this type of situation it is necessary to use synthetic models.

Synthetic models: These models attempt to realistically represent the behaviors of MNs without the use of traces, i.e., the previous record of data.

Entity Mobility Models: These mobility models represent the mobile nodes, whose movements are dependent of each other.

Our interest in this thesis is to analyze mobility models for MNs. Such, a mobility model should attempt to mimic the movements of real MNs. Changes in the speed and direction must occur as they would in reasonable time slots. For example, we would not want MNs to travel in straight lines at constant speeds throughout the course of the entire simulation if the real MNs would not travel in such a restricted manner. Below are seven different synthetic entity mobility models used for MANET [6]:

1. Random Walk Mobility Model, including its many derivatives is a simple mobility model based on random directions and speeds.
2. Random Waypoint Mobility Model is a model that includes pause times between changes in destination and speed.
3. Random Direction Mobility Model forces the MNs to travel to the edge of the simulation area before changing direction and speed.
4. Gauss-Markov Mobility Model uses one tuning parameter to vary the degree of randomness in the mobility pattern.
5. A Probabilistic Version of the Random Walk Mobility Model utilizes a set of probabilities to determine the next position of an MN.

6. City Section Mobility Model consists of a simulation area that represents streets within a city.

7. Group mobility models in which the MNs’ decisions on movement depend upon the other MNs in the group. The group mobility models can be further divided as [6],
   - Exponential Correlated Random Mobility Model: A group mobility model that uses a motion function to create movements.
   - Column Mobility Model: A group mobility model where the set of MNs form a line and are uniformly moving forward in a particular direction.
   - Nomadic Community Mobility Model: A group mobility model where a set of MNs move together from one location to another.
   - Pursue Mobility Model: A group mobility model where a set of MNs follow a given target.
   - Reference Point Group Mobility Model: A group mobility model where group movements are based upon the path traveled by a logical center.

This thesis focuses on the models that have been proposed for (or used in) the performance evaluation of a MANET. Details of these Mobility Models are discussed in the following sections.

3.2 RANDOM WALK MOBILITY MODEL

The RWMM was first described mathematically by Einstein in 1926 [7]. Since many entities in nature move in extremely unpredictable ways, the RWMM was developed to mimic this erratic movement [8]. In this mobility model, an MN moves from its current location to a new location by randomly choosing a direction and speed in which to travel. The new speed and direction are both chosen from the pre-defined ranges, \([speed_{min}; speed_{max}]\) and \([0;2\pi]\) respectively. Each movement in the RWMM occurs is based on time or distance of travel, where a new direction and speed are chosen in constant intervals. If a MN which moves according to this model reaches a simulation boundary, it “bounces” off the simulation border with an angle determined by the incoming direction. The MN then continues along this new path.

Many derivatives of the RWMM have been developed including the 1-D, 2-D, 3-D, and d-D walks. The 2-D RWMM is of special interest, since the Earth’s surface is modeled
using a 2-D representation. Figure 3.1 shows an example of the movement observed from this 2-D model.

![Figure 3.1. Travel pattern of an MN using the 2-D Random Walk Mobility Model.]

Few drawbacks of RWMM are: it is a memory less mobility pattern because it retains no knowledge concerning its past locations and speed values [9]. The current speed and direction of an MN is independent of its past speed and direction [10]. This characteristic can generate unrealistic movements such as sudden stops, and sharp turns, and also collision between nodes in the simulation area.

### 3.3 Random Waypoint Mobility Model (RWPMM)

The RWPMM is the most commonly used mobility model in research community [11]. In the current network simulator (ns-2) distribution, the implementation of this mobility model is as follows: at every instant, a node randomly chooses a destination and moves towards it with a velocity chosen randomly from 0 to the maximum speed chosen. After reaching the destination, the node pauses for a predetermined. After this duration, it again chooses a random destination and repeats the whole process again until the simulation ends (Figure 3.2).

The RWPMM acts as the 'baseline' mobility model to evaluate the protocols for MANET. The RWPMM is provided by the “setdest” tool in NS-2.
3.3.1 Common Problem with RWPMM

**Zig-zag trajectories:** RWPMM is elementary and it is easy to argue about the paths being unnatural. Then again, any practical protocol or mechanism should be robust and give a reasonable performance with a wide range of moving patterns, including movement similar to RWP model.

3.3.2 RWPMM with Attraction Points

Sometimes it is desired to model a ‘hot spot’ in a simulation, i.e., a certain area in which many nodes are located. Since the resulting node distribution of the RWPMM (Figure 3.2) is independent of the starting values of the nodes (for long simulation times), we cannot create such a hot spot by placing many nodes in a certain area at the beginning of the simulation, as is possible with other mobility models. Actually, the RWP model automatically creates a hot spot in the middle of the simulation area.

However, we can easily create an ‘attraction area’ anywhere in the simulation area by using a non-homogeneous distribution of the destination points [12]. When randomly selecting a destination point, a node chooses a point in this area with a higher probability than a point outside this area.

3.4 **Random Direction Mobility Model**

The Random Direction Mobility Model (RDMM) was created to overcome ‘density waves’ in the average number of neighbors produced by the RWMM [13]. A density wave is the clustering of nodes in one part of the simulation area. In the case of the RWMM, this clustering occurs near the center of the simulation area. In the RWMM, the probability of an
MN choosing a new destination that is located in the center of the simulation area, or a destination which requires travel through the middle of the simulation area, is high (Figure 3.2). Thus, the MNs appear to converge, disperse, and converge again. In order to overcome this type of behavior and spread the MNs throughout the simulation area, the RDMM was developed. In this model, MNs choose a random direction in which they to travel similar to the RWMM. An MN then travels to the border of the simulation area in that direction. Once the simulation boundary is reached, the MN pauses for a specified time, chooses another angular direction (between 0 and 180 degrees) and continues the process.

A slight modification to the RDMM is the Modified Random Direction Mobility Model. In this modified version, MNs continue to choose random directions but they are no longer forced to travel to the simulation boundary before stopping to change direction. Instead, an MN chooses a random direction and selects a destination anywhere along that direction of travel. The MN then pauses at this destination before choosing a new random direction. This modification to the RDMM produces movement patterns that could be simulated by the RWMM with pause times. As we can observe from the Figure 3.3, the MN’s are not concentrated in center, but spread over the entire simulation area.
3.5 Gauss-Markov Mobility Model

The Gauss-Markov Mobility Model (GMMM) was originally proposed for the simulation of a personal communication system (PCS) [14]; however, this model has also been used for the simulation of a MANET. The GMMM was designed to adapt to different levels of randomness via one tuning parameter. Initially each MN is assigned a current speed and direction. In fixed intervals each MN updates its speed and direction. At each time interval the next location is calculated based on the current location, speed, and direction of movement.

To ensure that an MN does not remain near an edge of the grid for a long period of time, the MNs are forced away from an edge when they move within a certain distance of the edge.

As shown in Figure 3.4, the GMMM can eliminate the sudden stops and sharp turns encountered in the RWMM by allowing past velocities (and directions) to influence future velocities (and directions).

![Figure 3.4. Travelling pattern of a node using the Gauss-Markov Mobility Model.](image)

3.6 A Probabilistic Version of Random Walk

This mobility model utilizes a probability matrix to determine the position of a particular MN in the next time step [6], which is represented by three different states for $x$-axis and three different states for position $y$-axis. In Figure 3.5, state 0 represents the current position of a given MN, state 1 represents the MN’s previous position, and state 2 represents the MN’s next position if the MN continues to move in the same direction. The probability matrix used is:
Figure 3.5. Flow chart of the probabilistic version of Random Walk.

\[ P = \begin{pmatrix} P(0;0) & P(0;1) & P(0;2) \\ P(1;0) & P(1;1) & P(1;2) \\ P(2;0) & P(2;1) & P(2;2) \end{pmatrix} \]

\( P(x;y) \): Probability that an MN will go from state \( x \) to state \( y \).

The values within this matrix are used for updates to both the MN’s \( x \) and \( y \) position. With the values defined, an MN may take a step in any of the four possible directions (i.e., north, south, east, or west) as long as it continues to move (i.e., no pause time). In addition, the probability of the MN continuing to follow the same direction is higher than the probability of the MN changing directions. Lastly, the values defined prohibit movements between the previous and next positions without passing through the current location. This implementation produces probabilistic rather than purely random movements (Figure 3.6), which may yield more realistic behaviors.

Figure 3.6. Example of traveling pattern of an MN using the probabilistic version of the Random Walk Mobility Model.
3.7 City Section Mobility Model

In the City Section Mobility Model (CSMM), the simulation area is a street network that represents a section of a city where the MANET exists [15]. The streets and speed limits on the streets are based on the type of city being simulated. For example, the streets may form a grid in the downtown area of the city with a high-speed highway near the border of the simulation area to represent a loop around the city. Each MN begins the simulation at a defined point on some street. An MN then randomly chooses a destination, also represented by a point on some street. The movement algorithm from the current destination to the new destination locates a path corresponding to the shortest travel time between the two points. Upon reaching the destination, the MN pauses for a specified time and then randomly chooses another destination (i.e., a point on some street) and repeats the process.

Figure 3.7 shows the movements of an MN using an example city section in CSMM. This model provides realistic movements for a section of a city since it severely restricts the traveling behavior of MNs. In other words, all MNs must follow predefined paths and behavior guidelines (e.g., traffic laws). In the real world, MNs do not have the ability to roam freely without regard to obstacles and traffic regulations.

![Figure 3.7. Traveling pattern of an MN using a probabilistic version of Random Walk.](image)

In addition, people typically tend to travel in similar patterns when driving across town or walking across campus. Enforcing that all MNs follow predefined paths will increase the average hop count in the simulations compared to other mobility models.

In the following sections we will be discussing group mobility and related models.
3.8 Group Mobility Models

In a MANET there are many situations where it is necessary to model the behavior of MNs as they move together. For example, a group of soldiers in an emergency scenario may be assigned the task of searching a particular plot of land in order to destroy land mines, capture enemy attackers, or simply work together in a cooperative manner to accomplish a common goal. In order to model such situations, a group mobility model is needed to simulate this cooperative characteristic. In this section, we discuss few group mobility models.

3.8.1 Column Mobility Model

The Column Mobility Model (CMM) proves useful for scanning or searching purposes. This model represents a set of MNs that move around a given line (or column), which is moving in a forward direction (e.g., a row of soldiers marching together towards their enemy). A slight modification of the CMM allows the individual MNs to follow one another (e.g., a group of young children walking in a single line to their classroom). Figure 3.8 shows the implementation of this model, an initial reference grid (forming a column of MNs) is defined. Each MN is then placed in relation to its reference point in the reference grid; the MN is then allowed to move randomly around its reference point via an entity mobility model.

Figure 3.8. Movements of three MNs using the Column Mobility Model.
3.8.2 Nomadic Community Mobility Model

Just as ancient nomadic societies moved from one location to another, the Nomadic Community Mobility Model (NCMM) represents groups of MNs that collectively move from one point to another [6]. Within each community or group of MNs, individuals maintain their own personal “spaces” where they move in random ways. Numerous applications exist for this type of scenario. For example, consider a class of students touring an art museum. The class would move from one location to another together; however, the students within the class would roam around a particular location individually. Figure 3.9, shows an example of such model.

![Figure 3.9. Movements of seven MNs using the Nomadic Community Mobility Model.](image)

3.8.3 Pursue Mobility Model

As the name implies, the Pursue Mobility Model (PMM) attempts to represent MNs tracking a particular target [16]. For example, this model could represent police officers attempting to catch an escaped criminal. The current position of an MN and acceleration of the MN are combined to calculate the next position of the MN. Figure 3.10 gives an illustration of six MNs moving with the PMM. The white node represents the node being pursued and the solid black nodes represent the pursuing nodes. Again, a simulated movement pattern for the PMM could easily be generated using the implementation of the RPGMM.
3.8.4 Reference Point Group Mobility Model

The RPGMM represents the random motion of a group of MNs as well as the random motion of each individual MN within the group. Group movements are based upon the path traveled by a logical center for the group. The motion of the group center completely characterizes the movement of its corresponding group of MNs, including their direction and speed. Individual MNs randomly move about their own pre-defined reference points, whose movements depend on the group movement [10].

Figure 3.11 gives an illustration of three MNs moving with the RPGM model. At time $t$, three black dots represent the reference points, $RP(t)$, for the three MNs. As shown, the RPGM model uses a group motion vector $GM$ to calculate each MN’s new reference point, $RP(t+1)$, at time $t+1$; as stated, $GM$ may be randomly chosen or predefined. The new position for each MN is then calculated by summing a random motion vector, $RM$, with the new reference point. Movement patterns using the RPGM model are shown in Figure 3.11. Figure 3.12 is an illustration of five groups moving, such that each group has a different number of MNs.

The RPGM model was designed to depict scenarios such as an avalanche rescue. During an avalanche rescue, the responding team consisting of human and canine members work cooperatively. The human guides tend to set a general path for the dogs to follow, since they usually know the approximate location of victims. The dogs each create their own “random” paths around the general area chosen by their human counterparts.
3.9 MOBILITY MODEL FOR EMERGENCY SCENARIOS

During battlefield planning, topographical teams and support staff are responsible for conducting thorough terrain analyses to support commanders in battlefield planning. This analysis can range from elevation calculations and specifications of restricted and unrestricted terrain, to soil and vegetation data depending upon the specific needs of the commander and the battle situation.

Each vehicle, or in some cases each soldier, represents a node in a larger tactical Internet.

On the battlefield, RPGMM is very useful. As, military units are fundamentally hierarchical, and they deploy, move and operate in groups that display tight adherence to a group structure that is known \textit{a priori}. Many other application scenarios, such as a fleet of warships or fighter planes in a combat maneuver, can also be modeled using RPGMM. As such, all nodes will move within the area based on the random waypoint mobility model.
3.9.1 Virtual Track Based Group Mobility Model

The key idea of this model is to use “virtual tracks” to model the dynamics of group mobility [17]. Some “switch stations” are first randomly deployed in the field. These stations are then connected via virtual tracks with given track width. The grouped nodes must move following the constraint of the tracks. At the switch stations, a group can then be split into multiple smaller groups; some groups may be even merged into a bigger group. Such group dynamics happen randomly under the control of configured split and merge probabilities. Nodes in the same group move along the same track. They also share the same group movement towards the next switch station. In addition, each group member will also have an internal random mobility within the scope of a group. The mobility speeds of these groups are randomly selected between the configured minimum and maximum mobility speeds. One can also define multiple classes of mobile nodes, such as pedestrians, cars, UGVs, and UAVs, etc. Each class of nodes has different requirements: such as moving speed etc. In such cases, only nodes belonging to the same class can merge into a group. Figure 3.13 illustrates a main idea of the virtual track based group mobility model. In this example, five switch stations are randomly placed in the field connected via eight virtual tracks with equal track width. Nodes in groups are moving towards switch stations along the tracks. They split and merge at switch stations as shown in Figure 3.13.

![Figure 3.13. Overview of Virtual Track Based Group Mobility Model.](image)

3.9.2 A Flocking-Based Mobility Model

In this model is for operations where a high degree of self-organization is required on a low level [18]. Self-organization on the micro level must go hand in hand with the ability to focus on a common task. The behavior of shoals and herds of land animals is fascinating in its seemingly unpredictable, yet coordinated motion. The dynamics resemble some of the key
features above. This makes models of flocking interesting candidates for simulation of the types of conflict that we are interested in.

Reynolds model is based on an “individual unit hypothesis”, meaning that the units act according to a set of rules that are applied in their neighborhoods. In addition to being governed by the external state, i.e. the neighbors in the vicinity, the particles internal state also govern the motion. Here, the internal state of a particle is just its velocity. The particle neighborhood is defined by two parameters, \( \theta \) and \( r \), see Figure 3.14.

![Figure 3.14. The neighborhood of a unit is defined by the angle and the distance \( r \).](image)

To participate in a flock each unit strives to stay close to the flock while avoiding collision with other units. Stated simply, the rules that determine the movement of a unit are (see Figure 3.15):

1. Avoid collisions with nearby units,
2. Attempt to match velocity with nearby units, and
3. Attempt to stay close to nearby unit

![Figure 3.15. Flocking rules.](image)
3.10 MOBILITY MODELS FOR UAV APPLICATIONS

Future applications of MANET are envisioned to cover a variety of scenarios in which networks of mobile entities cooperate in collecting, processing, and transmitting data in large geographical areas. An UAV added to the ground embedded mobile backbone can naturally form a multi-level physical heterogeneous multi-hop network, which is the best infrastructure for multi-area emergency environments. In the following sections we address the specific requirements of unmanned aerial vehicles (UAVs) for MANETs [19], cooperating to achieve a common mission. The aim is to model an abstraction of realistic emergency/surveillance mission scenarios.

3.10.1 UAV Movement

A fixed wing aircraft is limited in its movement in that it has a minimum and maximum air speed and that an instant change of direction is not possible. As we are mainly interested in the behavior of the system of UAVs a brief description of the movements of the individual UAVs has been used. The UAVs’ movements are limited to fixed speed, constant radius turns, and no collisions.

3.10.2 Hover above Movement UAV Protocol

This protocol determines the positions of the UAVs [20], which aim for flying above the nodes on the ground, is motivated by swarm intelligence techniques. In a swarm intelligence protocol, every node decides autonomously, i.e., without any communication with other nodes, about its movements. The only information that is required is the distance to nearby nodes and obstacles. In ‘Hover above’ movement protocol, an UAV determines its next targeted position based on the current positions of other UAVs and nearby nodes on the ground. Figure 3.16 shows a snapshot of the simulation with three UAVs hovering above the platoon at an altitude of 50m. The radio propagation between the ground nodes is calculated by the ray optical propagation model [21], and the communication between ground and UAVs is calculated by the Free-Space model under the consideration of buildings as obstacles is used.
3.10.3 Landmark Routing in Large Wireless Battlefield Networks Using UAVs

MANETs have great advantage and importance in the emergency environments because of its independence of the infrastructure and its instant deployment and easy reconfiguration capabilities. However, large-scale MANETs have performance limitations. The main reason being “long hop” paths, heavy routing overhead, and routing information that will become stale due to mobility of nodes. Thus, it is natural that the “flat” architecture in Figure 3.17 cannot fully support the MANET environment where a very large-scale network is needed; especially in emergency scenarios. Building a hierarchical MANET is a good way to solve this performance bottleneck.

Landmark Ad Hoc Routing (LANMAR) model [22], implements a hierarchical structure shown in Figure 3.18. The ordinary ground nodes with limited short transmission range are divided into groups. Each group has one backbone node. These backbone nodes have an additional, powerful radio and can form a backbone network with UAVs in the sky. LANMAR achieves scalability in large networks using a hierarchical routing scheme. LANMAR is very efficient in large, mobile, MANETs.
Routing information from UAVs to remote nodes is summarized by landmarks. Landmarks act as mediators between UAVs and the ground nodes to pass the routing information to other groups in the mission. This will greatly reduce the number of hops.

From Figure 3.19, we can see that by utilizing the hierarchical links, the six hop path is reduced to three hops, a great improvement. Here for simplicity the landmark and backbone node is the same node. However, this is not generally necessary.

### 3.10.4 Semi-Random Circular Movement Model

The SRCMM model is the only mobility model for generating traces for circular mobility of UAVs [23]. Circular mobility pattern are suitable for simulating UAVs hovering over specific location to gather information. UAVs usually tend to be assigned a specific destination according to mission requirements, and then follow a well-defined path to reach
the destination. Once they reach the destination they start hovering in circular fashion to collect the information and communicate the information to the nearest hop. SRCM model takes all this into consideration when generating the movement pattern. It assumes that UAVs have autonomous navigation with collision avoidance systems, which means no ground based navigation is required.

The advantages of SRCM model are: it can depict a curved/circular movement scenario which is a common movement pattern for UAVs. Secondly, UAV movements have adaptability to adjust their movement parameters to dynamic targets realistically.

### 3.10.5 Mobility Models for UAV Group Reconnaissance Applications

In this model UAV decides on its actions in small intervals of time according to Table 3.1 [19]. If the UAV moves outside the search area then it turns towards the centre of the search area until it has reached a randomly chosen direction between -45° to 45° w.r.t the edge of the search area.

<table>
<thead>
<tr>
<th>Table 3.1. UAV Random Action Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last action</td>
</tr>
<tr>
<td>Straight ahead</td>
</tr>
<tr>
<td>Turn left</td>
</tr>
<tr>
<td>Turn right</td>
</tr>
</tbody>
</table>

The pheromone repel model is used as basis for this mobility model that is robust and random [24]. Hence, each UAV maintains its own pheromone map. The pheromone map is a grid with element size 100*100 meters where each element contains a timestamp representing the last time the element was scanned. As a UAV moves, it marks the areas that it scans on the map. To share this information with the other UAVs, each UAV regularly broadcasts (every 10 seconds) a local area pheromone map (a square 5000*5000 meters centered at its current position). All UAVs within the broadcast range merge this information into their pheromone map. The reason for not broadcasting more frequently and only broadcasting a local map is to limit the size of data transferred over the wireless medium. As
with the random model a UAV decides to turn left or right or go straight ahead every other second, as shown in Table 3.2. But instead of making this decision with fixed probabilities, the probabilities are based on the pheromone smell in three areas as shown in Figure 3.20. Since a UAV should go to places not recently visited it should prefer areas with a low pheromone smell. Left, center and right are the pheromone smells of the areas in Figure 3.20 and total is their sum. If no pheromone smell is reported for any direction then a random direction is chosen as in the random model. If the center and either the left or right has no smell then a random direction is chosen between these two. The area outside the search area is given a high pheromone smell for the UAVs to avoid it. A special rule has been added to handle the case when a UAV flies directly into a corner of the area.

<table>
<thead>
<tr>
<th>Probability of action</th>
<th>Turn left</th>
<th>Straight ahead</th>
<th>Turn right</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(Total – Left) / (2 * Total)</td>
<td>(Total – Center) / (2 * Total)</td>
<td>(Total – Right) / (2 * Total)</td>
</tr>
</tbody>
</table>

**Table 3.2. UAV Pheromone Action Table**

![Figure 3.20. Pheromone search pattern.](image)

### 3.11 Motivation for a New Mobility Model Generator

Simulation results will greatly affect the routing protocol being used. The results can be unreliable if the model does not mimic the desired node movements. Therefore it is important to evaluate a routing protocol with the mobility model that most closely matches the expected real-world scenario.
In typical emergency scenarios, more complex mobility behavior is observed. Some nodes move in groups, while others move individually and independently, and a fraction of nodes are static. There can be UAV surveillance also involved in such scenarios. Moreover, the groups formed are not always permanent. The mobile groups can split or merge between themselves based on the scenario. All these different mobility behaviors coexist in emergency scenarios. The existing mobility model generators do not take all these factors into consideration. So we have designed a new mobility model generator which can represent the node movements in different scenarios, such as:

1. Emergency area mobility model
2. Multiple emergency areas with group mobility
3. Mobility Model for UAV Surveillance
4. Circular motion for UAV Surveillance and Emergency operations
5. Elliptical Mobility Model for UAV Surveillance

The scenarios mentioned above are discussed in the following sections and are generated using a mobility generator developed during my research work. This was done using Perl, NS-Nam. The Appendix has all the details of the code that has been developed in this project.

### 3.11.1 Emergency Area Mobility Model

In military scenarios the troops start from local station (marked (1) in Figure 3.21) and move towards a destination (marked (3) in Figure 3.21). Occasional reload stations are also represented in this model (marked (2) in Figure 3.21). While operating at the current emergency area, if a new emergency area comes up (marked (4) in Figure 3.21) the troops move towards the second emergency scenario.

**Scenario:** In Figure 3.21 we show the mobility pattern generated for the mentioned scenario. The troops initially start from origin (0,0) at 0sec towards an emergency area (3) and reach this emergency area to analyze the situation in group (snapshot taken at 49.3 sec of simulation). Once this emergency area is taken care of, during the simulation the nodes can move from one emergency scenario to another. Hence the nodes move towards new emergency area at 500 sec simulation time. The other attraction point (2) is used for occasional armor reload and food supplies.

### 3.11.2 Multiple Emergency Areas with Group Mobility

This is an extension of the previous mobility model, where we only considered individual nodes moving towards the target destination. In this model we have incorporated
group mobility as most of the military troops on ground form small groups and move towards their target destination. If there are two emergency destinations, the military troops generally divide into two groups to take care of these missions. So in Figure 3.22 below we simulate the scenario where military groups take care of two different emergency scenarios in groups.

**Scenario:** This model uses group mobility model, and the scenario in Figure 3.22 depicts 10 groups of five nodes each which start towards two different destinations (marked (2) and (1) in Figure 3.22 (a)).

In Figure 3.22(b) the six groups reach destination (2) and four groups reach destination (1) at simulation time 507 sec. In this model, we again consider multiple
emergency scenarios. The previous model, in such scenario sends all the troops available to the new destination. But here we divide the troops already at destination (2) to go towards the newly informed destination (3) in (c). Thereby addressing the issues at both the destinations based on the requirement.

3.11.3 Mobility Model for UAV Surveillance

Most of the emergency scenarios require the UAVs hovering over the area for surveillance and emergency response. UAVs can be used in surveillance missions to analyze the situation on ground and use this information to help the ground troops. Using this information the ground troops can go where their help is anticipated. The following model discusses the mobility model for UAVs to cover an area for surveillance during emergency scenarios.

The algorithm is chosen in such a way that, no UAV collides with each other and no UAV goes out of range of any other UAV in the network, so that they can relay information to command and control station. The UAVs start from a starting point as shown in Figure 3.23 (a) and steer towards the designated surveillance area as shown in Figure 3.23 (b). Once they reach the destination they start hovering to scan the area as shown in Figure 3.23 (c). The UAVs stay in their area of coverage. Once the UAVs reach their assigned destination they can follow different patterns to hover over the emergency area. So we discuss two patterns for UAVs to hover over an emergency area. The hover patterns of the UAVs decide the amount of information gathered from an emergency scenario. UAVs in their assigned area can hover in spiral or elliptical orbits, these motions are discussed in the following sections.

3.11.4 Circular Motion for UAV Surveillance and Emergency Operations

Once the UAVs reach their assigned emergency area they start hovering over the emergency scenario. RWP mobility model cannot be used for such hovering scenarios because of random sharp turns in irregular directions, which is not the way UAVs hover. Hence we came up with a circular mobility for UAVs to hover over an emergency area. The advantage of circular mobility model is that the scanned area under the UAV is more and it is uniform when compared to RWP model.
Figure 3.23. UAV surveillance mobility model (a) UAVs start toward their assigned emergency areas. (b) UAVs steering towards their assigned emergency area’s (c) UAVs hovering over the assigned emergency areas.

**Scenario:** We have simulated two nodes, node 0 and node 1 in Figure 3.24. Here node 1 is stationery throughout the simulation scenario and node 0 moves in circular pattern in anti-clockwise direction. This can be understood from the snapshot of the simulation in Figures 3.24 (a) (b) (c) and (d). The only disadvantage of the circular mobility is that if the area that needs to be covered is square/rectangle in shape. The edges cannot be covered. But this model can be easily extended to spiral motion to cover the desired locations easily.

Figure 3.24. (a)(b)(c)(d) Collectively shows circular motion of Node ‘0’.

### 3.11.5 Elliptical Mobility Model for UAV Surveillance

As discussed the disadvantage with the circular model was the area covered under the circular mobility model is always the same and the corners are never covered. Hence we designed an elliptical mobility model where the corners are covered easily by changing the major and minor axis length when needed. There is no mobility models based on elliptical orbits for UAV surveillance scenarios. Practically UAVs hover in elliptical orbits in surveillance for improving the connectivity of ground nodes. So, modeling this behavior and testing it would be a great advantage for future missions.

**Scenario:** In Figure 3.25 we chose 12 nodes (0-11), out of which the nodes 0 to 5 move in clockwise direction and 6 to 11 move in anti-clockwise direction in elliptical orbits.
Figure 3.25. (a)(b)(c) Depict the elliptical mobility of nodes (0-5) in clockwise direction and nodes (6-11) in anti-clockwise direction.
CHAPTER 4

QUEUING MECHANISM

In this chapter, we present a QoS-aware queuing scheme based on relative importance of video data in a transmission system. The importance of the video streaming packets is determined by three criteria: the frame number in a group of pictures (GOP), the H.264/AVC data packet priority [25]. Important packets of a video stream are given more channel access than the less important packets based on the available channel bandwidth. We demonstrate the effectiveness of this approach by simulations using NS2 and present a comparative study.

In following sections we will be reviewing few schemes which have been proposed earlier for optimizing transmission quality of multimedia data over wireless networks and the buffer/queue management schemes proposed to achieve QoS for multimedia data. We will later see how we use this information to come up with a better scheme for multimedia transmission over MANETs.

4.1 RELATED WORK

A large amount of work has been carried out to improve the quality of multimedia over wireless networks [26]. Existing wireless networks provide dynamically varying resources with only limited support for the QoS required by delay-sensitive, bandwidth-intense and loss-tolerant multimedia applications. One of the key challenges for delivering the multimedia data over wireless networks is the dynamic characteristics of both the wireless channels (varying supported bandwidth) and the multimedia source data. To overcome this challenge, packet scheduling and queue management schemes have been extensively investigated in recent years in order to maximize the quality of the multimedia streaming applications.

Various active queue management algorithms have been proposed for streaming multimedia in networks. The most widely used queue management algorithms are Drop Tail (DT) [27] and Random Early Drop (RED) [28].

**Tail Drop**: In this scheme every packet is treated as equally important. Once the queues are filled, the incoming packets are dropped until the queue is freed. The
method suffers due to alternative periods of empty and full queue which leads to bursty losses which is not desirable in video applications.

**Random Early Detection (RED):** In this method gateway detects congestion happening in the network and adapts itself by dropping packets or notifying the client through header files. But, it does not work when running on individual queues in wireless nodes.

The basic queue management methods discussed above might improve the throughput of the video being transmitted. However, the quality of video at the client is not directly related to throughput achieved. As discussed in the previous sections, it mainly depends on the frame type (I or P or B) from which the packet comes from and its priority. Hence we need smarter buffer/queue management schemes where these issues can be addressed. We will discuss few such schemes in this section.

Seong [29] proposed a video streaming framework that allows applications to mark packets with different priority and use multi-queue congestion control inside routers to effectively drop the less-important packets during buffer overflow. In [30] author proposed a buffer management scheme called Frame-Level Packet Discard with Dynamic Thresholds (FDDT), in which the packets are sorted/dropped based on the following conditions: (1) The first packet of B or P-frame is discarded when the buffer size reaches certain threshold levels; (2) An I-frame packet is only discarded when buffer is completely full; (3) If early packet of a frame is dropped, all the subsequent packets are dropped; (4) When two packets are competing, lower priority packets are dropped; (5) All incoming packets are dropped when the buffer is full. FDDT scheme showed results with great advantage for video quality, while having a small increase in computational complexity.

Shujie Wei [31] proposed an active queue management scheme based on bandwidth estimation for real time multimedia streaming over MANETS. The GM(1,1) model [32] was used to predict network bandwidth for determining the queue length, which was adjusted dynamically. In [33], a Drop Dependency Based (DDB) scheme was proposed, where basic information of the packet priorities was provided in the packet header and a buffer management was done based on this information. An optimized strategy operates on the Head of line (HOL: The packet which resides longest in the buffer) group of packets. By dropping the HOL packet with the lowest priority, a significant improvement in the video quality was achieved. This was extended in [34] to achieve an optimal combination of scheduler and drop strategy.
There have been some smart router based solutions such as Active networking [35]. In which routers play very important role of smartly discarding the packets based on \textit{a priori} knowledge of the transmission in progress. But, having functionality and computational complexity in the routers can go against the general norm of just forwarding the packets to the next node. But, this feature can be of great advantage in MANETs where the mission/video information of the current communication can be transmitted in the header during session establishment. In [36] author introduces a new term called fairness of service, which means the fair allocation of available resources to achieve the expected quality. It means that more resources are allocated for higher QoS requirement streams. The lower QoS requirement stream packets are dropped while giving more access to important streams. This can lead to starvation of resources for the later.

In Van der Schaar [37], a joint APP and MAC adaptation scheme was proposed with the use of MPEG-4 and its Fine Granularity Scalability (FGS) extension. In this work, packets containing multimedia data are classified into different classes and in the light of poor network conditions only packets with high class value are transmitted. The network conditions are jointly measured by combining the information obtained by the retransmission number of lost MAC frames (ARQ) and the information provided by the RTCP protocol.

Authors in [38] proposed a selective retransmission scheme for multimedia transmission over wireless networks. The idea was to retransmit only the important information in a video in order to achieve high quality of video streaming. Retransmission of important packets is a good idea, but since prevention is better than cure and we need to protect the high priority information and reduce the packet loss rate (PLR).

All the schemes discussed above, except [29], [30], mostly discuss the QoS of video by prioritizing different flows or by scheduling the higher priority packets by the application layer. But, there is very little research done for MANET environment on considering the relative priority of the video packets. This can play an important role in determining the video quality at the client node. Hence, based on the above discussions we propose a new queue management scheme which considers the relative priority of the video packets for real-time multimedia applications over MANETs.
4.2 H264/AVC Video Packet Prioritization

In this section, we consider the challenges of multimedia transmission over MANETs and introduce a new queue management strategy at the network layer to improve the H.264 video quality over MANETs. It is widely known that slice losses have different visual impact on the video. The transmission of H.264 video over the networks, which is agnostic of slice/macroblock priority, results in degradation of video quality due to poor allocation of resources to important slices of the video. The primary objective of this thesis is to use the prioritized H.264 video for transmission over MANET to give higher priority slices more channel access by pushing them ahead of the lower priority packets. The important slices are chosen based on the Cumulative Mean Square Error (CMSE) contributed by their loss. The proposed queuing strategy also considers other factors for sorting the video slices such as frame number and timestamp of the packets. We also drop the least priority packets if the queue length reaches an anticipated threshold level thereby dropping the packets at the queue itself, rather than dropping them in the network due to packet expiration. These factors will be discussed in detail in later.

As the priority is decided based on its impact on the decoded video we give more access to higher priority. Our queue management strategy makes sure that the lower priority packets won’t sit in the queue to delay the higher priority packets. A comparative analysis of the proposed scheme and the traditional queuing strategy is carried out. It is observed that the proposed scheme performs better than the traditional approaches.

4.3 Queuing Schemes in NS2

Queues represent locations where packets may be held/buffered (or dropped) until the routing requests find a route to the specified destination. Also, at the intermediate nodes the packets get buffered until the MAC layer gets channel access. Packet scheduling refers to the decision process used to choose which packets should be serviced or dropped. Buffer management refers to any particular discipline used to regulate the occupancy of a particular queue. NS-2 supports the drop-tail (FIFO) queuing, RED buffer management, CBQ (including a priority and round-robin scheduler), and variants of Fair Queuing including, Fair Queuing (FQ), Stochastic Fair Queuing (SFQ), and Deficit Round-Robin (DRR) [39].
NS2 implements PriQueue as a priority queue which gives priority to routing protocol packets, inserting them at the head of the queue. It supports running a filter over all packets in the queue and fetch the packet with a specified destination address. This makes sure that the routing packets reach the intended destination in minimum time. NS2 incorporates DropTail/FIFO (First In First Out) queue scheduling scheme and drop-on-overflow buffer management typical of most present-day Internet routers. This logic is implemented at the IFqueue shown in Figure 4.1 of NS2 architecture. We will be comparing the results of the proposed scheme with this scheme.

**Figure 4.1. Queuing scheme in NS2.**

### 4.3.1 InterFace Queue

IFQ is the queue which implements FIFO/DropTail mechanism, as shown in Figure 4.2, which acts as an interface between the link layer and the MAC layer. The packets coming from the link layer are queued in the IFQ and then when MAC has the channel access for the first packet in the queue it is transmitted. The interface queue accepts only two priority levels - the control packet is given the highest priority, i.e., whenever a control packet is received it is pushed to the head of the queue. All the data packets are treated with a lower priority as compared to the control packet. The easier way to tackle this issue is to have a single queue and employ a sorting algorithm, which classifies the packets based on priority
and sends out higher priority packets to the physical layer before it deals with the lower priority packets. This idea is the basis of this thesis work.

4.3.2 Event Scheduler

To drive the execution of the simulation and to process and schedule simulation events, NS-2 makes use of the concept of discrete event schedulers [40]. In NS-2, the network components that simulate packet-handling delay or that need timers use event schedulers.

Figure 4.3 shows two network objects, each using an event scheduler. If a network object issues an event, it has also to handle the event later at scheduled time. Based on the above discussion on the architecture of NS2, we will be discussing few drawbacks of the queuing scheme implemented with respect to multimedia transmission.

4.3 Priority Issue

At the application layer H.264 video packets are attached to the NS2’s Agent using Evalvid [41]. These packets are generated based on the H.264 trace file using the CMSE based prioritization scheme discussed above. The application layer creates a video packet and passes it to the lower layers. At the lower layers the packets are served as they arrive at the
queue in Network/MAC layer, regardless of their priority [42]. The general concept proposed in these schemes is that real-time packets are loss insensitive but delay sensitive. This means that packets should be served rapidly by the switch scheduler so that they get to their destination in the shortest time possible, even if some of them are lost. This reduces the need for buffering at the client. But, this is not quite true for multimedia data where packets have relative importance within them. Following such principle, the scheduling of packets in the conventional distributed coordination function (DCF) of 802.11MAC is FIFO based. The queue snapshot at a given time will be a set of packets waiting in the queue, without considering their priority. This effect might delay highly important slices and thereby the important video slices may not reach the decoder in time [43]. This will lead to inefficient decoding of video frames.

Let us understand this using an example. Consider priorities of video slices from p1-p4 with decreasing order of importance, where p1 is the most important packet and has to be given highest privileges for the channel access in the network. In Figure 4.4, we show a snapshot of video packets when no queue management is considered. As we can see the packet p1 is staying behind the comparatively less important packets p2 and p3. The higher priority packets are getting delayed in the queue. If the number of hops increases, the Packet Delay\(^1\) keeps increasing in the queues as it has to be served after the lower priority packets at

---

\(^1\) Packet Delay: Time taken by a data packet to be received at the receiver. This includes propagation, transmission, queuing and processing delay during the transmission of the packet.
each hop during the transmission. As the delay reaches a certain threshold value for a given video packet, the packet will be unfit to be used for decoding the frame. As a result, the packets have to be eventually dropped at intermediate nodes once the delay value exceeds a certain minimum threshold or the time-to-live.

### 4.4 Time Stamp Issue

Synchronized video is crucial for data acquisition and telecommunication applications. For real-time applications, out-of-sync video may cause jitter, choppiness and latency [44]. Timestamp control is required in order to show each picture at the right time with the right duration.

We must make sure that the delayed packet is dropped preemptively than transmitting it over the network and wasting the resources later.

In the given Figure 4.5 the packet with priority P1 and Frame number, fr1 has a timestamp of ts1, which has already exceeded the TTL limit assigned for this packet. But, the other packets have are still alive in the network. Even though the packet is priority p1 and is important for decoding of the video. It is not tolerable for the multimedia streaming application to have this packet in the network. It is treated as a dead packet by the decoder. It is a good choice to check TTL of the packet at every hop for every packet and if the limit is exceeded the packet have to be dropped.

### 4.5 Queue Overflow

FIFO or drop tail was the original queue management scheme used in Internet routers. With this scheme packets are en-queued at the tail of a queue as they arrive and de-queued from the head of queue when there is capacity on the link. Drop tail is the policy of dropping
the arriving packets when the queue is full. (Other alternatives include dropping the packet at
the head of the queue.) This strategy can lead to a major disaster in the network when there
are many flows in the network and only limited bandwidth is available. There are certain
situations where the queue condition cannot be reverted back to normal operation, if the
network continues to insert/send packets at the rate more than what the queue can handle [35,
36, 45].

Let us understand the situation in a queue with an example. The given queue has
video packets waiting to be transmitted to their respective receivers. In Figure 4.6 the queue
has reached its threshold limit and cannot insert any more packets. The packet P1 shown in
the diagram is waiting to be en-queued into the network. Since the queue is already full, this
packet has to be dropped. The effect of this depends on the type of the packet being dropped.
This will not affect background traffic as this information can be retransmitted at a later time
correctly. But for multimedia traffic if we lose a stream of five packets in a row due to queue
overflow the outcome can be disastrous for a video transmission.

Figure 4.6. Queue overflow.
Therefore care must be taken to keep track of the queue level and take actions before the queue level is reaching the maximum limit.

4.6 Lockout
This refers to a phenomenon in which the link bandwidth is unfairly consumed exclusively by a selected small number of flows [36]. The remaining flows are locked out to access the queue and do not receive any link bandwidth, which starve for resources.

4.7 Proposed Queuing Scheme for H.264 Video
Based on the above network conditions this thesis presents/proposes a set of algorithms for supporting QoS of multimedia traffic over MANET. The main objective is to present a new queuing logic at the network layer of the protocol stack to try and overcome the above-mentioned drawbacks of a FIFO/DropTail queuing.

4.7.1 At Enque
Here we discuss a few queue management strategies that have been implemented.

4.7.1.1 Check for Time to Live (TTL) while En-Queuing the Packet
During our simulation we set predefined TTL values for different priority packets. At any given instant if there is a packet in the network which exceeds this limit. We try to discard these packets at the earliest so that it will not use up any unwanted resources of the network.

When a packet is received after it has expired, it is dropped at the decoder even if the packet is received successfully. Therefore we drop the packet at the intermediate node if the time taken by it to reach the node has already exceeded its TTL.

We consider 4 different priorities for the video packets based on their CMSE and TTL values for each of these priorities p1, p2, p3, p4 is TTL_p1, TTL_p2, TTL_p3, TTL_p4 respectively.

Enque the packet: If the packet, as shown in Figure 4.7, to be enqueued is within TTL limit, then that packet has to be enqueued to transmit over the wireless network. For storing a packet at intermediate/source node, the following steps have to be considered.
Figure 4.7. Queue condition while en-queuing a packet.

\[
P_{\text{enque\_time}} = \text{Packet delay of the enqueing packet (so far)}
\]

\[
\text{for (i=1; i<=4; i++)}
\]

\[
\{ \text{if } (\text{packet\_prio} == p[i])
\]

\[
\text{if } (P_{\text{enque\_time}} > \text{TTL}\_p[i])
\]

\[
\text{Drop packet;}
\]

\[
\text{else}
\]

\[
\text{Enqueue packet}
\]

The node then waits for the next packet to be enqueued.

**4.7.1.2 Check for Buffer Fullness**

Once the enqueing is done, the packet will be waiting for it to be given a chance by the MAC layer to be transmitted over the network. If queue is about to reach its threshold upper limit, we drop lower priority packets based on percentage of queue fullness. This is
done in the following way, when a packet is ready to be enqueued, and then we check for the following conditions.

\[ Q_{\text{len}} = \text{Present queue length} \]
\[ Q_{\text{lim}} = \text{Maximum limit of the queue} \]

\[
\text{If}(Q_{\text{len}} > 80\% \text{ of } Q_{\text{lim}}) \\
\quad \text{Drop the priority p4 packets above half size of queue.}
\]
\[
\text{If}(Q_{\text{len}} > 90\% \text{ of } Q_{\text{lim}}) \\
\quad \text{Drop all the priority p4 packets in the queue.}
\]
\[
\text{If}(Q_{\text{len}} > 95\% \text{ of } Q_{\text{lim}}) \\
\quad \text{Drop all the priority p4 packets and half of the priority p3 packets in the queue.}
\]

The objective here is to drop less important packets to make sure the more important packets of the video stream have enough queue length/space to enque the packets. So that if the conditions are not favorable for the transmission of video data can be decoded at the receiver.

**4.7.2 Queue Sorting**

When a packet is within the TTL limit w.r.t its priority, we insert the packet in the queue in such a way that relatively important packets are given more chance of getting serviced over the other packets in the queue, as shown in Figure 4.8. The following things are considered before enqueing the packet.

**4.7.2.1 Frame Number Based Sorting**

In this section we introduce the algorithm for sorting the packets based on their frame number. As discussed frame number is based on the transmission order of the frames. Frame number of a packet is an important parameter that has to be considered for a video transmission (based on the results we observed). The idea here is to sort the packets of a given priority based on the frame number of the packets.

\[ \text{F\_current} = \text{Frame number of the current packet considered} \]
\[ \text{F\_enque} = \text{Frame number of the packet to be enqueued} \]
4.7.2.2 PRIORITY BASED SORTING

As discussed in the section Priority issue the video transmission has to be sorted based on the priority of the video packets.

Sorting based on priority:

$P_{current} =$ Priority of current packet considered in the queue
$P_{head} =$ Priority of the head of the queue
$P_{enqueue} =$ Priority of packet to be enqueued
Tail =$ Last packet in the queue
\[ P_{\text{current}} = P_{\text{head}} \] // we start from the head of the queue and start comparing
// with \( P_{\text{enqueue}} \) until end of the queue

Before sorting the packets based on the priority of the packets, traverse to the packet where the \( F_{\text{current}} \) is equal to the \( F_{\text{enqueue}} \) (From section Frame Sorting)

For \( (i=0; i<= 4; i++) \)

\{
  If (\( P_{\text{enqueue}} == p[i] \))
    If (\( P_{\text{current}} == P_{\text{head}} \))
      Insert packet before \( P_{\text{head}} \)
    Else
      Insert packet before \( P_{\text{current}} \)
    Else if \( P_{\text{current}} \) is Tail
      Insert the packet as tail
\}

This algorithm makes sure that all the higher priority packets are given higher channel access over the other lesser priority packets. Thereby we make sure the most important packets are received at the decoder so that the frame drops due to these packets can be reduced by a great extent. The results for this are shown in the later sections.

**4.8 AT DEQUEUE**

When the queue receives a “call_back” signal from the MAC layer to deque a packet. We perform similar kind of checks as performed at enqueue, before we deque the packet (Figure 4.9). In wireless networks there is a lot of interference and collisions in the Physical layer. This can lead to packets drops of video packets which can lead to a diminishing quality of the streaming video. So the parameters that we consider are the effect of network congestion/collisions over the video quality and current TTL values of each packet. The idea here is to detect the channel conditions at the earliest possible and then act accordingly to save the quality of the video to continue uninterrupted video.
4.8.1 Data Rate Control

Data rate control is basically to have check on the bandwidth that is supported by the wireless channel. Scalable video coding can be used to adapt in such varying conditions [36]. A simple way of it is to check if the network is unable to match the packet injection rate. This is tested by constantly monitoring the data rate of the network at intervals of GOP time of the video being transmitted. Then calculate the data rate supported by the network and if this is lesser than the encoder rate we try to drop the packets in the queue. This is a preemptive measure to make sure we drop packets before the queue is filled completely.

delta = GOP time for the considered video sequence.

\[ D = \text{The amount of data transmitted in ‘delta’ sec, in bytes} \]

Data rate of the network \( N = \frac{D}{\text{delta}} \)

\( V \) = Video encoding rate.

For every ‘delta’ Calculate \( N \)

If \( V > N \)

Drop all the remaining packets of previous GOP in the queue.

If \( V < N \)

Continue transmission after checking for TTL of the ‘head’ packet.
4.8.2 Check for TTL before Dequeueing a Packet

As discussed the packets get dropped at the decoder even if the packet is received successfully but if it exceeds the TTL limit for the packet. The TTL is checked even before enqueuing the packet. But, we want to make sure the packet which is has to be dequed to the MAC layer has enough time to be transmitted over the network.

The main objective of this logic is to drop the packets which are in the queue for longer duration and exceed their TTL limit. Such packets have to be dropped even while allowing them to be transmitted over the network. The counter argument can made stating that it takes very less time for a packet to reach the next hop and we anyways have a TTL logic implemented at the enque function. But for video streaming channel access plays a very critical role. The main idea here is to make sure that the resources provided by the MAC layer (which are scarcely available in a dense network) are utilized in an efficient manner. By dropping the packets preemptively we give a chance to packets which are hungry for resources.

The logic is implemented when MAC sends a callback signal to the IFQueue.

4.9 Results

In this sections, all the ideas discussed in the previous sections are implemented on NS2 for a H.264 video over MANET and thorough analysis is done for the results obtained. The following Table 4.1 shows scenario that has been used.

<table>
<thead>
<tr>
<th>Table 4.1. Network Scenario</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Number of nodes</strong></td>
</tr>
<tr>
<td><strong>TTL values</strong></td>
</tr>
<tr>
<td><strong>Queue length</strong></td>
</tr>
<tr>
<td><strong>Simulation area</strong></td>
</tr>
</tbody>
</table>

The simulation scenario chosen has 25 nodes in the network and the speed is varied in such a way that the results are obtained for business districts and highways. We haven’t considered walking speed (1-1.5m/s) as the results obtained were similar to a stationary network, for which MANETs are not applicable. The routing protocol chosen is according to
the scenario where at highway speeds there will be constant breakage of routes. AODV protocol will play a role of finding routes on Ad-Hoc basis.

In the following figures (Figure 4.10, Figure 4.11, Figure 4.12) we compare results for FIFO (NS2) logic with the current scheme discussed.

![Graph](image)

(a)

![Graph](image)

(b)

**Figure 4.10.** Packet delivery ratio using different priorities as a function of varying number of connections at average node speed of at $V = 5\text{m/s}$. Note that ns-2 uses a default drop-tail queuing mechanism which is priority-agnostic.
Figure 4.11. Packet delivery ratio using different priorities as a function of varying number of connections at average node speed of at $V = 10\text{m/s}$. Note that ns-2 uses a default drop-tail queuing mechanism which is priority-agnostic.
Figure 4.12. Packet delivery ratio using different priorities as a function of varying number of connections at average node speed of at $V = 10\text{m/s}$. Note that ns-2 uses a default drop-tail queuing mechanism which is priority-agnostic.
From the plots in Figure 4.10, Figure 4.11, Figure 4.12 it is clearly understood that the present queuing logic has better performance by giving higher channel access to more important packets. The following is the summary of the results.

1. As packets of a frame number $X$ are transmitted only after all the packets of the frame number $X-1$. Hence, though we are sorting packets w.r.t priority within a frame. There might be a chance of priority 1 packets of Frame number $X$ sitting behind the packets of lesser frame number $X-1$. In such case the following logic comes into picture.

   Encoder transmits packets without considering the prevailing network conditions. Usually in a video sequence for a given GOP last few frames are always lower priority packets. When a new GOP starts the first frame is an IDR frame. Since the transmission of packets is independent of network situation, these IDR packets sit behind the rest of the previous GOP packets. This situation is seen when the network is unable to cope up with encoder transmission rate or when there are severe packet drops due to collisions. And packets get accumulate in the queue.

   The logic of dropping packets by foreseeing the network conditions and preemptively discarding the packets of previous GOP makes sure the early packets of the next GOP which have most number of higher priority packets are given the best channel access. Even though we have TTL conditions for each priority packet, the above condition gives a significant advantage for higher priority packets. Because we make sure the network discards all the packets of previous GOP which no longer can reach the destination in time. So it directly gives a better chance for the higher priority packets of the next GOP which are new in the network to be transmitted in time. Because of this logic (along with other logic, discussed in following sections) we can see a significant improvement in the PDR values of priority p1 and p2 packets.

2. After sensing channel conditions and dropping unwanted packets in the network. We need to make the sure right packets are given the best chance to be transmitted. So within a given frame all the higher priority packets are sorted to the top of the queue and the rest follow in decreasing order of priority. This gives the higher priority packets more channel access and therefore better throughput.

3. When a packet is ready to be serviced by the MAC layer the conventional way of servicing is to serve the head of the queue. But what if this packet has already expired and is transmitted to the next hop; this will waste the available resources. As the delayed packets will be late for decoding. So we try to drop packets preemptively which are not suitable for decoding and service only the packets which can be used by the decoder.

   From the results obtained and the discussion above it is evident that all the logics that have been discussed so far work cumulatively to give higher throughput for important packets of a video transmission.
CHAPTER 5

CONCLUSION

In this thesis, we came up with a mobility model generator for NS2 for military scenarios. Various new scenarios were proposed using this mobility generator. This can be used to evaluate different routing protocols for a variety of military scenarios.

In the end we propose a scheme of improving the throughput of video over MANETs using sorting algorithm based on relative priority of packets and packet scheduling. The results have shown that the more important information packets can be given more channel access to improve the quality of video streaming.

This section presents some directions for future work. Possible avenues in which this work can be extended include the following.

1. **Source Priority:** As we have seen so far in the above discussion, we have only considered priority of the packets for a given source. But, in a real network scenario, we have multimedia and background data together. So the current algorithm can be extended to include the priority of the source nodes based on the information being transmitted.
   - **Implementation details:**
     - For a given packet in the network if source priority and priority of packet are as shown the priority of the packet is changed based on the combination:
       - Source1, priority 1  => priority 1
       - Source1, priority 2; Source2, priority  => priority 2
       - Source1, priority 3; Source2, priority 2  => priority 3
       - Source1, priority 4; Source2, priority 3  => priority 2
       - Source2, priority 4  => priority 5

2. The scenarios considered for the experiments in this thesis are based on RWPMM. It would be interesting see how the algorithm behaves for different scenarios as discussed in the Mobility Model Generator section.

3. Integration with MAC layer to know the channel condition (collisions, channel interference etc). Based on this information we can regulate the outgoing packets by dropping/sending packets based on the packet priority.

4. This Algorithm can also be tested with in-house LAM-AOMDV protocol, to check the behavior of the scheme in highly mobile network where path breakages are more.
REFERENCES


APPENDIX

MOBILITY MODEL GENERATOR
Mobility Model Generator:

Scope: This document is a detailed explanation of how to generate the mobility model script for circular and elliptical orbits.

Files include:

- Positions.pl,
- Node_n_co_orondiate.pl,
- UAV_circular_movement.pl

1) **Position.pl:**
   
   **Usage:** Positions.pl x_grid<>y_grid<>node_speed<>simulation_time<> radius <> PIC <num of positions in one circle(a common multiple of 4 & 3)
   
   X_grid: The breadth of the Grid.
   Y_grid: The length of the Grid.
   Node_speed: Speed of the node.
   Simulation_time: Time of the simulation
   Radius: Radius of the circle.
   PIC: number of points in the circle. (This must be a common multiple of 4 & 3. Ex: 12)dfa

   This code execution generates positions that a node has to take in order to move in Circular orbit for the specified time (Simulation_time). This has to be taken into a file called positions.out. We use this file to replicate this motion for other nodes in the simulation scenario.
   
   Note: Positions.out file generates files starting from node(0),(0,0) co-ordinates, time – 0sec.

2) **Node_n_co_ordinate.pl:**
   
   **Usage:** Node_n_Co-ordinate.pl <nodes-offset><x-offset><y-offset><Movement-trace-filename><time-offset>

   **nodes-offset:** This is a tricky number. This has to decided carefully, This can be best explained with an example.
   
   If the scenario we chose consists of 30 nodes with mobility pattern as follows:
   (0-9)10 nodes at 20m/s in circular orbits
   (10-19) 10 nodes at 5 m/s using RWP model
(20-19) 10 nodes at 5 m/s using GMM
   Note: For group mobility “Bonnmotion generator” has to be used to generate Positions.out file.

For such a scenario we generate positions.out file for the various scenarios chosen and we have to apply this scenario for the respective node. So node_offset is ‘0’ for circular orbit model, is ‘10’ for RWPM, is ‘20’ for GMM

**x-offset, y-offset:** These parameters are similar to node-offset, because “positions.out” print positions w.r.t origin. If nodes have to start at (100,200) we chose the offset as 100 for x-axis and 200 for y-axis.

**Movement-trace-filename:** this is the positions.out file generated for a given simulation scenario.

**time-offset:** this is similar too, but if we want to start the mobility of nodes at time other than ‘0’, ex: 100sec. we give it that value.

**Result:** The Excecution of this file generates a file of format “Emergencyscenario-nnode-offset-xx-offset-yy-offset-ttime-offset.out” this has the movement values for the nodes from the offset value chosen starting from the offset position at time-offset.

**Note:** Make sure you create a directory called results where the out files are stored, which are used by third script.

3) **UAV_circular_movement.pl:**
   This file is an automation script for Node_n_co_ordinate.pl, which runs this script for the number of nodes chosen and uses the “Emergencyscenario-nnode-offset-xx-offset-yy-offset-ttime-offset.out” file of each node to generate one file of format “12345Emergencyscenario_UAV.out”. This file has the movement file for all the nodes for the scenario chosen compatible with NS2 simulations.

**Result:** A file of “setdest” style for node movement during simulation scenario.