Bandwidth Management and Security in Mobile Ad Hoc Networks

Benny Munyaradzi Nyambo

Promoters
Prof. Dr. Gerrit K. Janssens
Prof. Dr. Wim Lamotte

A Thesis Submitted in fulfilment of the Degree of
PhD in Computer Science

Hasselt University
2017

Acknowledgement

My gratitude and appreciation go to my promoter, Prof. dr. Gerrit Janssens for his guidance throughout my doctoral program, his invaluable contribution to this research. I learnt a lot in writing clear and correct work from him. I will forever value this knowledge from him.

I wish to thank Prof. dr. Wim Lamotte for his willingness to be my co-promoter. I will always value his criticism and insight.

I wish to thank the Vlaamse Interuniversitaire Raad Universitaire Ontwikkelingssamenwerking (VLIR-UOS) for giving me a scholarship, to pursue a doctoral degree, and Hasselt University for granting me chance to study at this beautiful university.

I wish to thank Ms. Martine Dekoninck (Dept. International Development Cooperation, KULeuven) and the International Office staff at Hasselt, Ms. Sarah Verlackt, Ms. Erika Vandersmissen and Mr. Marc Thoelen for assisting me with my visa application and accommodation every year, and the Finance Office and the Secretariat of the Faculty of Business Economics for assisting me with the financial and administrative tasks necessary during my doctoral program.

Finally, I would like to thank my wife Linda and my children Benjamin, Benedict and Blessing for their support during my study.

Benny Munyaradzi Nyambo
Diepenbeek, 2017

TABLE OF CONTENTS

1. Introduction 13

1.1. Motivation.......13

1.2. Purpose and Significance of Study 14

1.3. Methodology 15

1.4. Structure of the Research 18

2. QoS in MANETs 19

2.1 Introduction.......19

2.2 Quality of Service 21

2.3 Quality of Service Metrics 21

2.3.1 Throughput.......22

2.3.2 Network Delay 25

2.3.3 Delay variation/Jitter 27

2.3.4 QoS Requirements for Voice, Video, and Data 28

2.3.5 Human perception to QoS.......34

2.3.6 User Perceptive Quality of service.......35

2.4 Quality of Service Issues in MANETs.......37

2.5 Wireless Sensor Networks.......40

2.6 Smart phone Mobile ad Hoc networks.......42

2.6.1 WiFi Direct 42

2.6.2 Melon 43

2.6.3 Open Garden 44

2.6.4 Serval Project 45

2.6.5 SPAN 46

2.6.6 AdHocDroid 46

2.6.7 Comparison of smart phone MANET technologies 47

2.7 Quality of Service Models for MANETs.......48

2.7.1 Introduction 48

2.7.2 IntServ 48

2.7.3 Differentiated Service (DiffServ) 50

2.7.4 FQMM 52

2.7.5 INSIGNIA.......53

2.7.6 SWAN 57

2.7.7 ASAP 59

2.7.8 CEQMM 63

2.7.9 QoSMMANET.......64

2.7.10 iMAQ 65

2.7.11 DEQA 65
2.8 Conclusion 70

3. Bandwidth Estimation 72
3.1. Problem Definition 72
3.2. System definition 73
3.3. Bandwidth Estimation Techniques 75
3.3.1 Model based techniques 75
3.3.2 Measurement based approaches 82
3.3.3 Calculation-Based Approaches 92
3.4. Model Formulation 102
3.5. Experimentation and analysis 103
3.6. Results 105
3.7. Conclusion 110

4. Bandwidth Management Framework for MANETs 112
4.1 Problem definition 112
4.2 System Definition 113
4.2.1 Resource Estimation 114
4.2.2 Signalling 114
4.2.3 Admission Control 116
4.2.4 Resource Reservation 116
4.2.5 Adaptation 117
4.3 Model formulation 118
4.3.1 Traffic Differentiation 119
4.3.2 Admission Control 121
4.3.3 QoS Signalling Method 122
4.3.4 Re-routing (Fast restoration). 125
4.3.5 Bandwidth Adaptation 127
4.3.6 The Effect of adaptation on the codec 130
4.4 Experimentation and Analysis 131
4.5 Results and findings 133
4.5.1 No Adaptation and admission of flow 135
4.5.2 Adaptation and admission of flow 135
4.5.3 Adaptation and no admission of flow 136
4.5.4 Analysis of results 137
4.6 Conclusion 142

5. Time-Slot Assignment in TDMA MANETs 144
5.1 Problem definition 144

5.2 System definition 145
5.2.1 Formulation of problem as a graph 146
5.2.2 Time slot assignment and QoS routing 149
5.3 Model formulation 154
5.3.1 Prioritised Dining Philosophers Algorithm 156
5.3.2 PDP priorities assignment 157
5.4 Experimentation and Analysis 162
5.5 Results 163
5.6 Conclusion 168
6. Security in MANETs 170
6.1 Problem definition 170
6.2 System definition 171
6.2.1 Over reservation attack 171
6.2.2 State table starvation attack 171
6.2.3 QoS Degradation 172
6.2.4 Flooding attack 172
6.2.5 Replay Attack 173
6.3 Model formulation 174
6.3.1 Neighbour Verification 177
6.3.2 Localized Token Issuing Process 178
6.3.3 Trust manager and neighbour monitoring 181
6.3.4 Secure Path Finding 185
6.4 Experimentation and analysis 190
6.4.1 Performance Metrics 190
6.5 Results 192
6.5.1 Packet Delivery Ratio 192
6.5.2 Malicious node detection rate......**Error! Bookmark not defined.**
6.5.3 False positive ratio 195
6.5.4 Selective Packet Dropping Rate 196
6.5.5 Node Trust 197
6.6 Conclusion 199
7. Final conclusions 200
7.1 Introduction 200
7.2 Contributions of the thesis 200
7.3 Future Work 204
REFERENCES 206

APPENDIX A 214

APPENDIX B 216

APPENDIX C 216

APPENDIX D 218

List of Tables

Table 1: Performance targets for audio, video and data applications 31
Table 2: VoIP Bandwidth Reference Table 33
Table 3: VoIP Bandwidth Needs with Layer 2 Overhead 33
Table 4: Bit rates for audio streams 35
Table 5: Listening-quality scale 36
Table 6: Manets network solutions checklist ((Soares et al., 2017) 47
Table 7: QoS Table for ASAP 61
Table 8: Comparative analysis of QoS models in litererature 66
Table 9: Summary of Path / Bandwidth Analysis Tools 93
Table 10: Modified Hello Structure. 100
Table 11: The simulation results of the DBE for the first 10 seconds 105
Table 12: The variation of packet loss with the network-breaking rate 107
Table 13: Variation of available bandwidth with the number of nodes 108
Table 14: Variation of throughput against number of nodes in the network 109
Table 15: Cisco QoS Baseline Classification, Marking, and Mapping 120
Table 16: Hop-by-Hop Options extension header format 123
Table 17: QoS Table kept at each node keeps details for each flow 125
Table 18: Traffic classes used in the simulations. 132
Table 19: Load already in the network at the beginning of a simulation. 133
Table 20: Simulation results showing adaptation 134
Table 21: Sample results for admission without need for adaptation 135
Table 22: Sample results after adaptation and then admission 136
Table 23: Adaptation and no Admission 136
Table 24: Variation of nodes adapted downwards with iniitial network load 138
Table 25: Variation of admitted nodes with initial network load 140
Table 26: Variation of number of flows denied access with network load 141
Table 27: Slot status table maintained by every node 159
Table 28: Sample flows, priorities, sizes, and arrival times of traffic flows 162
Table 29: Parameter values 190
Table 30: Packet delivery ratio for DSR-S* compared to DSR 193
Table 31: Malicious node detection rate 194
Table 32: False positive detection rate 195
Table 33: Packet dropping rate 196
Table 34: Variation of node trust with time spent by a node in the network ... 198

Table of Figures

Figure 1: The general structure of the research 18

Figure 2: A mobile ad hoc network (MANET) (Xiao and K.G. Seah, 2000) 20

Figure 3: Simplified view of the access speed 23

Figure 4: Difference between achievable bit rate and access speed 25

Figure 5: Network transit delay 26

Figure 6: End to end delay in a network 26

Figure 7: Packet jitter caused by the network 27

Figure 8: End-user QoS categories mapping. (Rafael et al., 2007) 29

Figure 9: An illustration of Hidden terminal problem 39

Figure 10: An illustration of Exposed Node Problem 40

Figure 11: A sensor network used for prompt fire detection 41

Figure 12: The operation of RSVP 49

Figure 13: The DiffServ Architecture (Iqbal, 2008) 51

Figure 14: An illustration of type of nodes in FQMM depending on scenarios. ... 52

Figure 15: The INSIGNIA IP option in a packet. 54

Figure 16: Examples of INSIGNIA operations.(Studi, Xue, & Alonso, 2004) 55

Figure 17 ASAP soft reservation 59

Figure 18: ASAP hard reservation (Studi et al., 2004). 60

Figure 19: The QoSMMANET Framework. 64

Figure 20: An H-1 hop path to calculate available bandwidth 73

Figure 21: CSMA/CA protocol with RTS/CTS exchange mechanism 76

Figure 22: One step transition Markov Chain model (Bianchi, 2000) 77

Figure 23: Model based bandwidth estimation analytical model 80

Figure 24: One-way delay for two streams using SLoPs 83

Figure 25: The Probe Gap Model for estimating bandwidth. 84

Figure 26: Offered bandwidth over measured bandwidth in TOPP 88

Figure 27: Plot of the ratio o_i/m_i as a function of o_i. 89

Figure 28: The interference distance between a receiver (R) and sender (X) 94

Figure 29: An example of channel state at different times 95

Figure 30: The listening bandwidth estimation mechanism 97

Figure 31: RTS/CTS Access Scheme 99

Figure 32: Neighbour cache structure 101

Figure 33: The structure of the Dual bandwidth estimation method 102

Figure 34: The flow diagram of the Dual Bandwidth Estimation method 103

Figure 35: The simulated environment for available bandwidth of node *x* 104

Figure 36: Usage of the two methods in relation to network stability 106

Figure 37: Effect of network path breaking on pcket loss............................108
Figure 38: Variation of available bandwidth with the number of nodes 109
Figure 39: Variation of throughput with the number of nodes in the network .110
Figure 40: Traffic Management Framework Architecture 118
Figure 41: The functions of Differentiated Services (DS) bits 119
Figure 42: The function of the Admission Control Module 122
Figure 43: Signalling messages embedded in the IPv6 Header 123
Figure 44: Fast reservation showing request and reservation report 124
Figure 45: Fast restoration after a link is broken by due to mobility. 126
Figure 46: Bandwidth Adaptation module 128
Figure 47: Adaptation: scaling up in case of more resources availed 129
Figure 48: Adaptation: Scaling down in case of resources lost......................130
Figure 49: Admission success rate of adaptation Vs without adaptation.........138
Figure 50: The variation of number of flows reduced with load increase 139
Figure 51: Variation of admitted nodes with initial network load 140
Figure 52: Variation of nodes denied access with the network load 141
Figure 53: Number of flows against the consumed bandwidth. 142
Figure 54: Division of time into frames and time slots. 146
Figure 55: Illustration of nodes in conflict..147
Figure 56: TDMA frame is split into control phase the data phase slots 150
Figure 57: Path set up between a source S and Destination D 151
Figure 58: Re-routing have to be initiated after a path break 153
Figure 59: Petri net for dining philosophers' problem. (Mcguigan, 2007) 155
Figure 60: Variation of states in the PDP assignment algorithm 156
Figure 61: The DP problem for slot S_i, contended for by two nodes..............158
Figure 62: A Petri Net for the PDP model for two nodes in contention 159
Figure 63: Prioritised Dining philosophers algorithm flow diagram 160
Figure 64: Average cumulative waiting time for the control experiment 164
Figure 65: The waiting time for PDPA flows of various priorities 165
Figure 66: Throughput for flows for priority assignment for the control 166
Figure 67: Throughput for PDPA flows for priority assignment 167
Figure 68: The variation of average cumulative admissions with time 167
Figure 69: Composition of the proposed network layer security solution 177
Figure 70: Localized token issuing and message handshaking 179
Figure 71: Packet Delivery Ratio 193
Figure 72: Malicious Node Detection 194
Figure 73: False Positive Rate 195
Figure 74: Selective Packet Dropping Rate 197

Figure 75: Node Trust for DSR-S* over Time 198

Acronyms

AIMD	additive-increase multiplicative decrease
AMAN	Adaptive Mobile Ad Hoc Network
AODV	ad hoc on-demand distance vector
AP	access point
ARQ	automatic repeat request
ASAP	Adaptive ReServation and Pre-Allocation Protocol
BE	best effort
BER	bit error rate
CBR	constant bitrate
CDMA	code-division multiple access
CEQMM	Complete and efficient quality of service model for MANET
CLD	Cross Layer Design
CSI	channel-state information
CSMA/CA	carrier sense multiple access with collision avoidance
CSMA/CD	carrier sense multiple access with collision detection
CTS	clear to send
CW	contention window
DCF	distributed coordination function
DEQA	Design of an Efficient QoS Architecture
DHCP	Dynamic Host Configuration protocol
DiffServ	differentiated services
DIFS	DCF interframe space
DSCP	Differentiated service code point
DSR	Dynamic Source Routing
DSR-R*	Dynamic Source Routing with Security
FDMA	frequency-division multiple access
FQMM	Flexible Quality of Service Model for Mobile ad hoc networks
GO	Group Owner
IBSS	Independent Basic Service Set
IEEE:	Institute of Electrical and Electronics Engineers
iMAQ	Integrated Mobile Ad-hoc QoS framework

IntServ	integrated services
IP	Internet protocol
IPv4	Internet protocol Version 4
IPv6	Internet protocol Version 6
ITU-T	International Telecommunication Union
MAC	media access control
MANET	Mobile Ad Hoc Networks
MOS	mean opinion score
PER	packet error rate
PGM	Probe Gap Model
PHB	per-hop behaviour
QoE	quality of experience
QoS	quality of service
QoSMMANET	QoS Management in Mobile Ad hoc Networks
RED	random early detection
RIP	routing information protocol
RREP	Route Reply
RREQ	Route request
RSVP	resource reservation protocol
RTP	real-time transport protocol
RTS	request to send
SIFS	short interframe space
SIP	session initiation protocol
SPAN	Smart Phone Ad Hoc Network
SWAN	Stateless Wireless Ad hoc Networks
TCP	transmission control protocol
TOPP	Train of Packet Pairs
UDP	user datagram protocol
VBR	variable bitrate
VoIP	voice over IP
WBest	Wireless Bandwidth estimation tool
Wi-Fi	wireless fidelity

1. Introduction

1.1. Motivation

Mobile ad-hoc network (MANET) is an autonomous, dynamically reconfigurable wireless network without any centralized administration or infrastructure in which mobile nodes communicate directly and cooperatively with each other. Each device or node in the network has to take the responsibility of forwarding packets for its peers and a packet may traverse multiple nodes before it reaches the destination. Therefore nodes are able to operate as hosts and routers at the same time in a MANET. The IEEE has standardized IEEE 802.11 protocols to support MANET media access. There is great potential for the use of Mobile ad hoc networks in transmitting complex multimedia applications, where various Quality of Service (QoS) attributes for these applications must be satisfied as a set of predetermined service requirements. At a minimum, the QoS issues pertaining to throughput, delay, bandwidth management and security are of paramount interest. It is very important to find cost-effective ways of solving these issues at appropriate layers of the network for MANETs to find widespread use.

Since real time multimedia applications have intensive resource consumption, getting them to work over MANETs is a big challenge. MANETs have characteristics that bring challenges in guaranteeing quality of service. These include the known interference problems faced by wireless networks, limited bandwidth, multiple node functionalities, node mobility which leads to dynamic topologies, limited processing power, limited storage capacity, hidden and exposed terminal problems and many more other problems that may arise. All these challenges affect the provision of quality of service in mobile ad-hoc networks and influence greatly the issue of flow reservation in ad-hoc networks. It is of paramount importance to come up with methods of providing QoS in MANETs in view of all the challenges associated with them.

1.2. Purpose and Significance of Study

The primary aim of this research is coming up with a cross-layer quality of service framework for MANETs using simulation modelling. The objective study is twofold: (a) to describe the existing QoS frameworks in existence and finding their strengths and weaknesses and (b) to design a quality of service framework based on the strengths and weaknesses of the known frameworks.

Mobile ad hoc networks are expected to play a very important role in complex multimedia applications and communication. Multimedia applications have various minimum QoS requirements that have to be satisfied for them to be useful. Some of these requirements include, minimum delay, maximum throughput and effective bandwidth management. Mobile ad hoc networks are expected to be useful in the military, police, emergence services like fire department and other commercial applications around the globe. Therefore security issues in MANETs need to be addressed to ensure confidence of users.

MANETs have found use in various areas of life today, varying from include establishing survivable, efficient, dynamic communication for emergency/rescue operations, disaster relief efforts, and military networks, communication at the office, at home, in sensor networks, in inter-vehicular networks and even in the medical field. In some of the applications such as battlefield, medical field and vehicular networks, there is need for assurance of availability, accuracy and authenticity of data sent and received by nodes. Wrong or incomplete information can endanger human life in one way or the other. In emergency rescue missions and in battlefields, the data sent and received have to be trusted and received with limited delay and good throughput. The same argument can be used in vehicular ad hoc networks where vehicles have to send and receive information in order to increase the safety and comfort of the passengers.

This research is intended to help in discovering areas of concern in quality of service in MANETs and propose solutions in the form of a cross-layer QoS framework that is expected to be followed when implementing QoS in MANETs. These solutions are expected to go in a long way in contributing to the efforts of improving the quality of communication when MANETs are concerned. Other researchers have done work on QoS in MANETs and have come up with various QoS frameworks such as SWAN, INSIGNIA, ASAP and so on. In all this research, traffic is differentiated into two classes, real-time traffic and non-real-time traffic. However our research goes ahead and gives the users the room to put classes into real-time traffic since these also have different quality of service needs. We argue that internet game traffic cannot have the same priority with internet telephone. Video over IP requires more bandwidth than voice over IP so they cannot also have the same priority on the network. We propose a framework that classifies traffic according to its minimum quality of service requirements.

1.3. Methodology

For the greater part, QoS problems in MANETs are investigated according to the layer of the OSI model they are associated with. Algorithms are proposed to address these problems and simulations are done to investigate the suitability of each and every algorithm in comparison with existing algorithms or quality of service frameworks.

To solve the ultimate problem of supplying quality of service in MANETs we are going to look at methods that can help improve throughput, reduce delay and delay variations, reduce packet loss and ultimately providing security to the data in transit. This will be done mostly by managing the bandwidth and adapting to changes in available bandwidth in the network. This will be done by splitting the whole problem into sub-problems that can be solved individually at various network layers, but the solutions to these problems cooperatively give us the ultimate solution to providing quality of service in MANETs. These problems are bandwidth estimation, admission control and bandwidth management and finally network security.

The first problem that was solved is bandwidth estimation. This was the first problem to solve since every other effort to improve quality of service depends on knowledge of bandwidth available. Knowledge of available bandwidth is very vital in admission control and bandwidth management for the purpose of maintaining quality-of-service (QoS) in both wired and wireless networks. In wireless networks, the available bandwidth changes very fast with time because of channel fading, electromagnetic interference and errors caused by physical obstacles. The wireless network is a shared medium, in which all nodes in the same neighbourhood interfere and contend for the use of the channel. This makes available bandwidth estimation in a wireless networks environment, a difficult task. The ability to detect the link capacity and available bandwidth on wireless ad hoc networks is important for the success of multimedia applications such as videoconferencing. Real-time multimedia applications in mobile ad hoc networks would normally require prior knowledge of the both metrics in order to make admission control and bandwidth management decisions.

The second problem is to design a bandwidth management framework for MANETs. Projected applications of mobile ad hoc networks will require a distinction in the quality of connections being supported in terms of bandwidth availability, end-to-end delay and jitter. As multimedia traffic find its way into wireless networks, the use of UDP transport layer protocol alone would not suffice to the needs of users. As the number of UDP supported traffic increase the throughput of each flow is drastically reduced. Most multimedia traffic requires a more stable throughput for them to be regarded to be useful. Therefore there is a dire need for a QoS model integrated within the nodes for such applications. The model must be able to distinguish flows based on their QoS needs and have mechanisms that work to meet those requirements. Since there is no central coordinator, the QoS model must operate in a fully distributed manner. Such requirements combined with the hostile working conditions of an ad hoc network make the task of designing such a model very challenging. We propose a novel QoS model that differentiates the flows into classes and attempts to provide bandwidth and delay guarantees to flows of highest priority class. All this applies in MANETs that uses the IEEE802.11 MAC protocol, which relies mostly on CSMA/CD principle for channel access. However, not all MANETs use the CSMA/CD MAC protocol but others use TDMA as the underlying MAC layer protocol. This will lead us to the third problem.

The third problem is to come up with a time slot assignment solution in MANETs that use TDMA MAC protocol. We need to find a scheduling algorithm that will allow us to schedule traffic optimally by allocating time slots (schedules) to traffic such that there are few idle slots. We want the algorithm to take into account the different priorities in the traffics, giving higher priority traffic more slots. If the highest priority traffic requests the channel and if the channel is busy we want the algorithm to release the slots allocated to the least priority flow. This is an assignment problem. Assignment problems deal with the question how to assign *n* items (traffic) to *m* other items (machines tasks).Their underlying structure is an assignment which is nothing else than a bijective mapping between two finite sets of n elements.

The fourth problem that is solved is of security in MANETs. When the bandwidth, delay and throughput are good, the user wants to be assured that this data is secured. Most routing protocols assume a general trustworthy and cooperation of participating nodes. That is, they adopt a priori trust. This general assumption and the intrinsic nature of MANETs make routing protocols vulnerable to routing disruption attacks leading to Denial of Services (DoS) attacks. In this thesis we seek to come up with a set of efficient, trusted routing discovery and maintenance rules, and a security framework for establishing trust of cooperating nodes and this should make use of neighbour verification and monitoring techniques to enhance the Network-Layer security.

The fifth problem is the development of a cross layer framework that include individual QoS entities solved in the previous objectives. These include bandwidth estimation, bandwidth management, delay control and quality of service security. All these entities should be able to interact in a comprehensive fashion by sharing information across their respective layers without compromising quality of service. We would also investigate how our cross layer framework can work in a heterogeneous network. In this way the MANET will be connected to the internet through a gateway and we propose a signalling method that can work in such a scenario.

1.4. Structure of the Research

The structure of the research is shown if Figure 4. All the five problems being addressed are aimed fulfilling quality of service in mobile ad hoc networks.

By solving different problems associated with QoS in MANETs, we draw closer to achieving the ultimate goal of QoS in MANETs. In addressing these five problems it is hoped that appreciable good quality can be achieved in MANETs.

Chapter 2 describes fundamental properties of ad hoc networks and quality of service in mobile ad hoc networks. Chapter 3 deals with bandwidth estimation in IEEE802.11 MANETs and chapter 4 covers time slot assignment for QoS in a Time Division Multiple Access (TDMA) MANET. Chapter 5 is on designing a bandwidth management framework for MANET. Chapter 6 covers security in MANETs. Chapter 7 is a cross-layer design of quality of service in mobile ad hoc networks connected to the internet. Chapter 8 is for discussions and conclusions to the whole research of quality of service in mobile ad hoc networks.

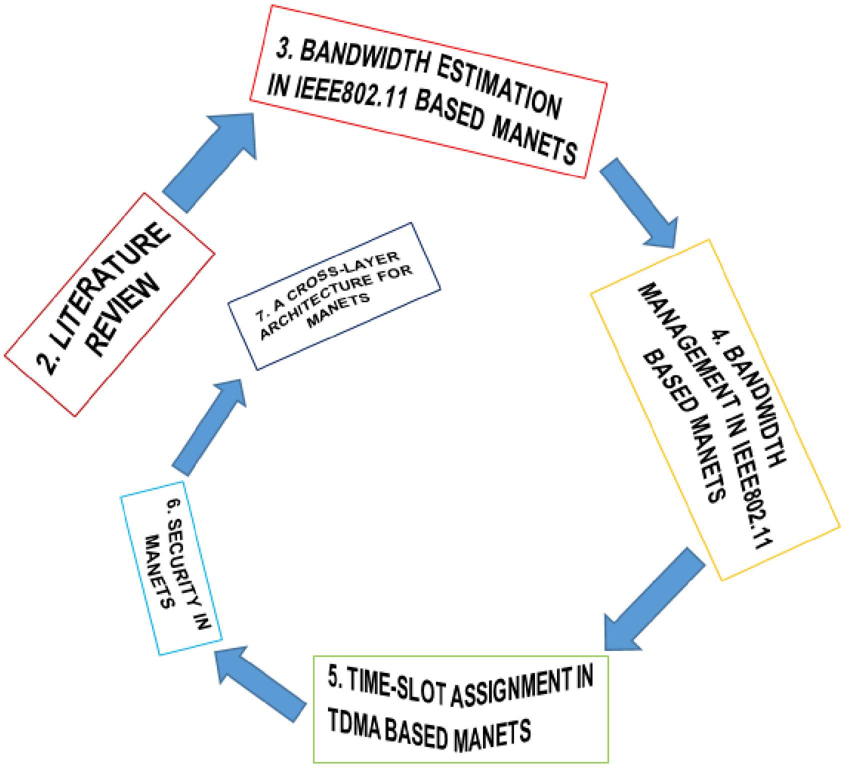

2. LITERATURE REVIEW
3. BANDWIDTH ESTIMATION IN IEEE802.11 BASED MANETS
4. BANDWIDTH MANAGEMENT IN IEEE802.11 BASED MANETS
5. TIME-SLOT ASSIGNMENT IN TDMA BASED MANETS
6. SECURITY IN MANETS
7. A CROSS-LAYER ARCHITECTURE FOR MANETS

Figure 1: The general structure of the research

2. QoS in MANETs[1]

2.1 Introduction

Wireless communication networking is one of the most significant technologies in the 20th Century (S. K, Sarkar et al 2008). Whilst this is an exciting development, for many people, multimedia is the holy grail of networking technologies. The former see immense technical challenges in providing (interactive) video on demand to every home. The latter see equally immense profits in it. This justifies a great need for research on wireless networks which carry multimedia traffic.

Wireless networks can be classified into two distinct groups: Infrastructured and infrastructure-less. Infrastructured networks are composed of mobile nodes, base stations and access points. The base stations and the access points form the core of the network and mostly they are fixed. All the routing information is stored in the core network and the host just need to pass information to the access point and the necessary route is found. In infrastructure-less wireless networks, there are only mobile nodes. Each node has to operate both as a host and a router. If a host receives information meant for another host it finds the best route to and forwards the information to the next host. The advantage of these networks is that, they are easy and cheap to set-up. They find potential use in areas such as tactical communication disaster response, battlefield, remote areas, sensor networks and many other scenarios that may arise from time to time. Infrastructure-less networks are also known as Ad hoc wireless networks. When coupled with mobility, they are called Mobile Ad hoc Networks (MANETs). In this research we will concentrate on properties and applications of MANETs only.

[1] The work in this chapter is an extension of work published in the International Journal of Information

In a MANET, nodes within interference range share status information so much that neighbours are conscious of the presence of all their neighbours. Mobile Ad hoc networks are becoming more and more popular with industry and individuals. MANETs are expected to become the future of wireless networks, because they are practical, versatile, easy to use and inexpensive to setup. We project a world where the network instantly updates and reconfigures itself to keep people connected wherever they go.

On the other hand there is a great advancement in multimedia transmission in networks. This has seen the emerging of Internet telephone or Voice over IP (VoIP), multimedia streaming and even real-time Video over IP. Because of all these improvements there is a demand for high speed wireless networks that are able to transmit data, audio and video traffic to ad hoc network users on the move (Ngatman, Ngadi, & Sharif, 2008). There may also be a need, in future, to hook up MANET users to the Internet. In this way a MANET would become part of a heterogeneous network. Although MANET users would require all the real-time services that wired network users enjoy there are still a lot of challenges that need to be addressed. Real-time traffic is much more sensitive to network quality of service (QoS) as compared to best-effort traffic such as email and file transfer. But MANETs are very low on bandwidth and they usually battery operated so they are power sensitive.

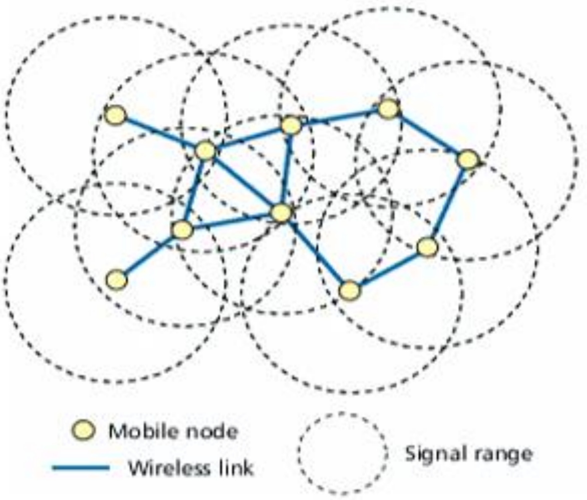


Figure 2: A mobile ad hoc network (MANET) (Xiao and K.G. Seah, 2000)

2.2 Quality of Service

Quality of Service (QoS) describes the level of user satisfaction of the services provided by a network, while transporting a packet stream from a source node to a destination node. In computer networks, the goal of QoS support is to achieve more predictable, reliable and deterministic behaviour, in order to preserve the information carried by the network and at the same time optimally utilizing the bandwidth. QoS can also be defined as the ability of a network element (e.g. an application, a host or a router) to deliver a high level data delivery performance beyond a prior expected value (Mohapatra, Li, & Chao, 2003). QoS is based on an agreement or a guarantee by the network to provide a set of measurable pre-specified service attributes to the user in terms of available bandwidth, probability of packet loss (loss rate), throughput, network delay, delay variance (jitter), and security. Different applications require different QoS requirements, from the network. Real-time applications are time sensitive and have to be delivered within an expected time delay, otherwise real-time communication would become impossible; non-real-time applications are concerned more on reliability instead. For multimedia traffic over the internet, the ultimate goal is to preserve both mission-critical data in the presence of multimedia voice and video whilst the quality of voice and video is maintained in the presence of busty data traffic. A significant amount of research has been done on the issue of QoS in MANETs over the years, however according to (Singh, Dutta, & Singh, 2012),(Anil Lamba, 2015), (Sobti, 2015), (Reina et al., 2015) and (Aarti & Tyagi, 2013) current results are not appropriate for MANETs such that QoS and security for MANETs remain an open problem. These survey researches give us a reason to look into the issue of QoS in MANETs closely. The issues of resource reservation and QoS make us conclude that, an adaptive QoS system needs to be implemented over the traditional resource reservation to support the multimedia services

2.3 Quality of Service Metrics

QoS metrics are base parameters of quality for a network. QoS parameters include throughput or bandwidth, delay, jitter, probability of packet loss or error rate, security, network availability, and battery life. The QoS could be defined in terms of the parameters or set of parameters in varied proportions (Mohapatra et al., 2003).

2.3.1 Throughput

Throughput or bit rate is the rate at which end-systems can exchange binary information. Bit rate and throughput are used interchangeably in industry and we are going to do the same in this thesis. Bit rate between two communicating end-systems is the number of binary digits that the network is capable of accepting and delivering per unit time. The unit for expressing throughput is the number of bits per second (bps), or bit rate. However the practical units are the kilobit (Kbps), the megabit (Mbps) and gigabit (Gbps).

Video and voice packets generally require large bandwidth; otherwise bottlenecks will develop in the network links leading to packet losses. Packet loss refers to the percentage of packets that fail to reach their destinations for various reasons. A packet loss of 1% produces a jerky video, while loss of 2% will start to render video unusable, though audio can be acceptable. Lost audio packets produce choppy, broken audio. Since audio operates with smaller packets at a lower bandwidth, in general, it is usually less likely to encounter packet loss, but an audio stream is not immune from the effects of packet loss. Packet loss in the 1-2% should still be considered a poor network environment and the cause of this type of consistent, significant packet loss should be resolved (Ngatman et al., 2008).

There are two notions associated with rates at the interface between an end-system and a network. These are the access speed (bandwidth) and the bit rates. The access speed is the frequency at which bits may be sent and received over the interface between the end-system and the network. This frequency is always determined by the technology used by the network. In certain cases this frequency is determined by independent clocking signals and bits can only be sent or received when matching these signals.

The available bandwidth of a path is a concave metric that defines the width of the path. In practice it is a bottleneck which defines the bandwidth that a service can be allocated to.

$$B_{avail} = min[B_x]_i^k \qquad 2.1$$

B_x is the bandwidth or the access rate at each node x in a given path from source *i* to a destination *k*. However, not all networks are capable of transporting data transmitted at the sustained access speed of the network interface. Several networks cannot accept data during certain periods because of internal congestion, lack of capacity, or because the user has subscribed to a bit rate lower than the access rate.

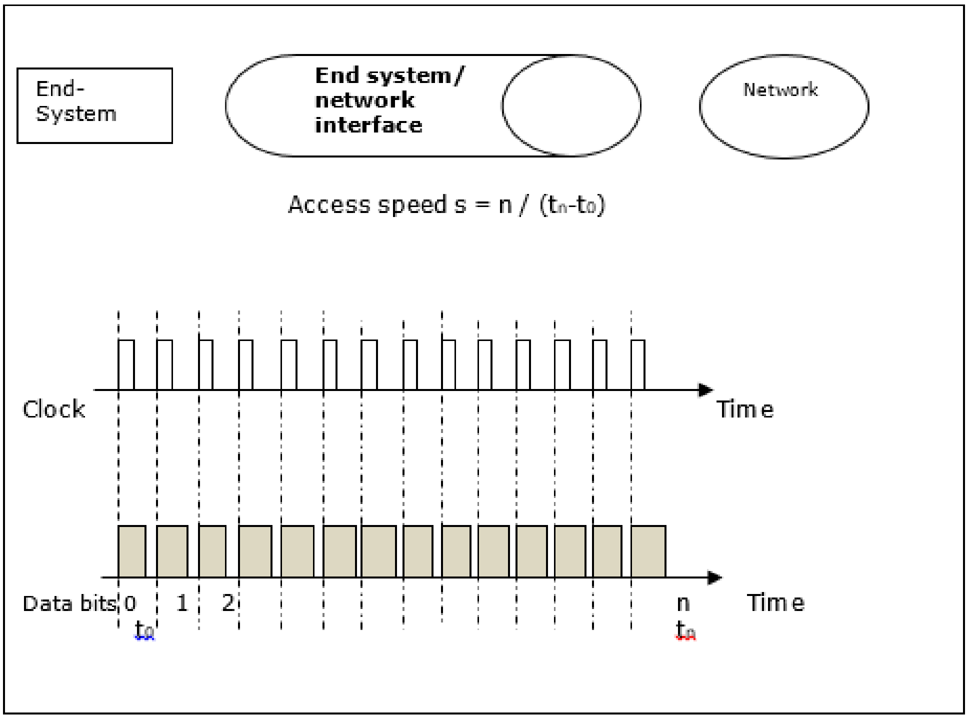


Figure 3: Simplified view of the access speed

In MANETs several factors will affect the overall throughput of any protocol operating in an ad hoc network. For example, node mobility sometimes may cause links to breaks thereby negatively affecting routing and overall QoS. When more nodes come into or leave the network, the amount of control overhead in the data, and the amount of traffic will have a considerable impact on network scalability. These factors coupled together with general characteristics of MANETs sometimes result in unpredictable variations in the overall network behaviour.

When data is transferred over a communications medium, such as a MANET, the average transfer speed is often described as throughput. This measurement includes all the protocol overhead information, such as packet headers and other data that is included in the transfer process. It also includes packets that are retransmitted because of network conflicts or errors. Goodput, on the other hand, only measures the throughput of the original data. Certain networks cannot accept a sustained traffic at access speed of the network interface. Goodput is the size of the transmitted data divided by the time it takes to transfer that data. Since this calculation does not include the additional information that is transferred between systems, the goodput measurement will always be less than or equal to the throughput. For example, the maximum transmission unit (MTU) of an Ethernet connection is 1,500 bytes. Therefore, any file over 1,500 bytes must be split into multiple packets. Each packet includes header information (typically 40 bytes), which adds to the total amount of data that needs to be transferred. Therefore, the goodput of an Ethernet connection will always be slightly less than the throughput.

While goodput is typically close to the throughput measurement, several factors can cause the goodput to decrease. For example, network congestion may cause data collisions, which requires packets to be resent. Many protocols also require acknowledgment that packets have been received on the other end, which adds additional overhead to the transfer process. Whenever more overhead is added to a data transfer, it will increase the difference between the throughput and the goodput.

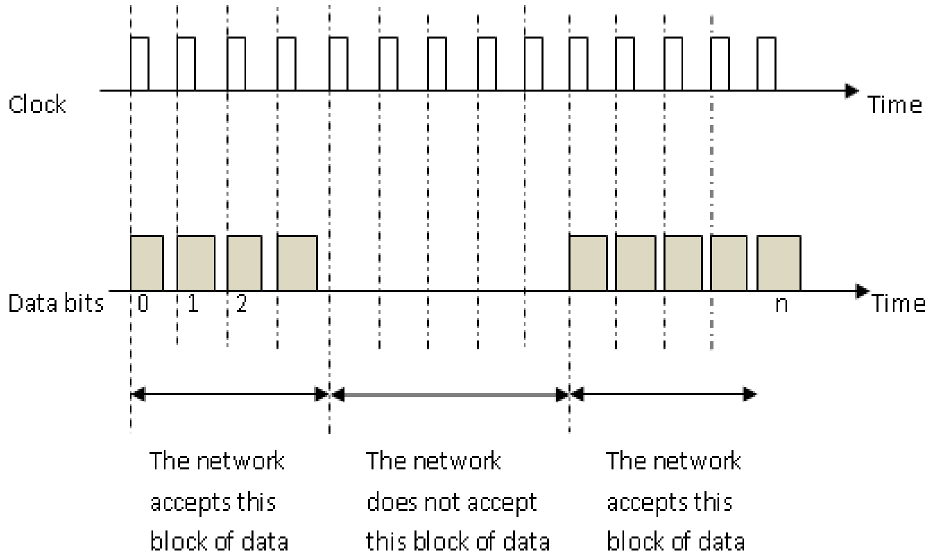

Clock
Time
Data bits
0
1
2
n
Time
The network accepts this block of data
The network does not accept this block of data
The network accepts this block of data

Figure 4: Difference between achievable bit rate and access speed.

2.3.2 Network Delay

Network latency or delay refers to the total transit time of packets to arrive at the remote endpoint. It is the time elapsing between the emission of the first bit of a data block by the transmitting end and its reception by the receiving end-system. No network can transmit a packet instantaneously, though certain networks have shorter latencies than others. Store-and–forward packet networks, based on packet switches or routers, may have substantial transit delays, up to seconds for long-haul connections. The total end-to-end delay consists of three components:

- Node processing and queuing delay, sometimes called access delay; this is the time spent at the source node waiting for the medium to be available in order for the network to be ready to accept the block of information.
- Transmission delay; this is the time taken by the node to actually transmit the sequence of bits of the blocks, one after the other, once the network is ready. It is a function of packet size and network bandwidth.

- Propagation or network transit delay; this is the actual time taken for packets to move between source node and destination node. It is a property of distance between the two nodes and propagation speed.

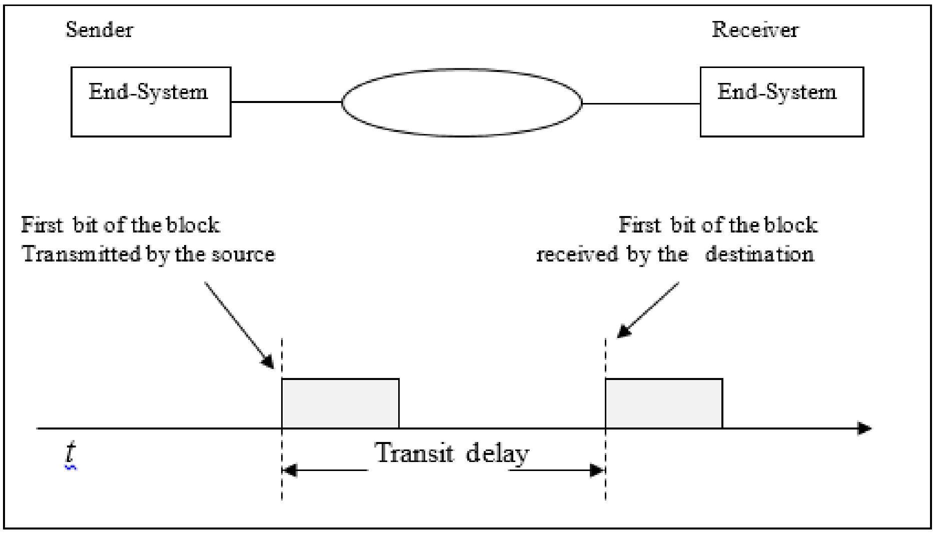


Figure 5: Network transit delay

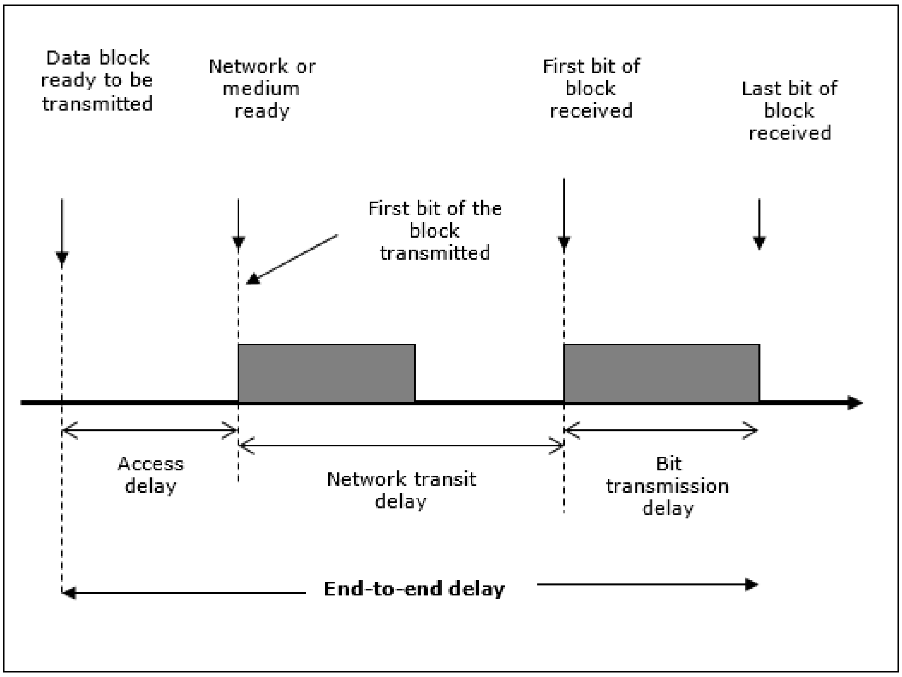

Data block ready to be transmitted
Network or medium ready
First bit of block received
Last bit of block received
First bit of the block transmitted
Access delay
Network transit delay
Bit transmission delay
End-to-end delay

Figure 6: End to end delay in a network

The end-to-end delay metric of a path is additive. It is the sum of the propagation delays of the path. It is also an indication of the length of the path. The propagation and queuing delays from a source of communication to the destination is additive. Suppose *d (i, j)* is the delay for link *(i,j).* The path p linking i to m nodes, *p=(i,j,k,...,l,m),* has delay D given by equation 2.2.

$$D = \sum_{x=i,y=j}^{x=l,y=m} d_{xy} \qquad 2.2$$

In equation 2.2, d_{xy} is the delay experienced in the link between nodes x and y. This means that effort has to be made to reduce delay in all links in a path from a source to the intended destination. Under the H323 protocol, delay should not exceed 125 - 150 milliseconds (Ngatman et al., 2008).

2.3.3 Delay variation/Jitter

Jitter is the variation in end-to-end delay for packets belonging to the same data stream. In transmission technology, jitter refers to the variation of the delay generated by the transmission equipment. It is generally caused by congestion in the network, either at the interfaces of routers or in a carrier network provided the circuit has not been designed in the proper way. Figure 7 illustrates jitter of packets traversing the internet.

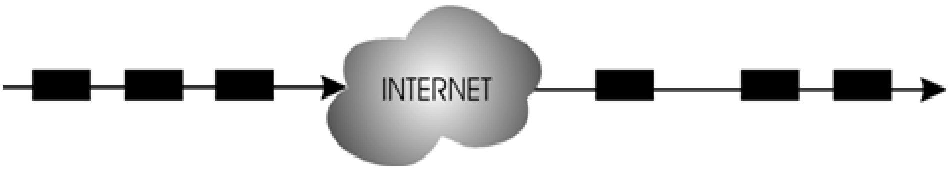


Figure 7: Packet jitter caused by the network

We can express average jitter q_j experienced as:

$$q_j = \frac{1}{n-1}\sum_{l=2}^{n} |(R_l - R_{l-1}) - (S_l - S_{l-1})| \qquad 2.3$$

where S_l is the time the packet *l* was sent from source, and R_l is the time at which packet *l* arrives to its destination (Veres, Campbell, & Barry, 2001) .

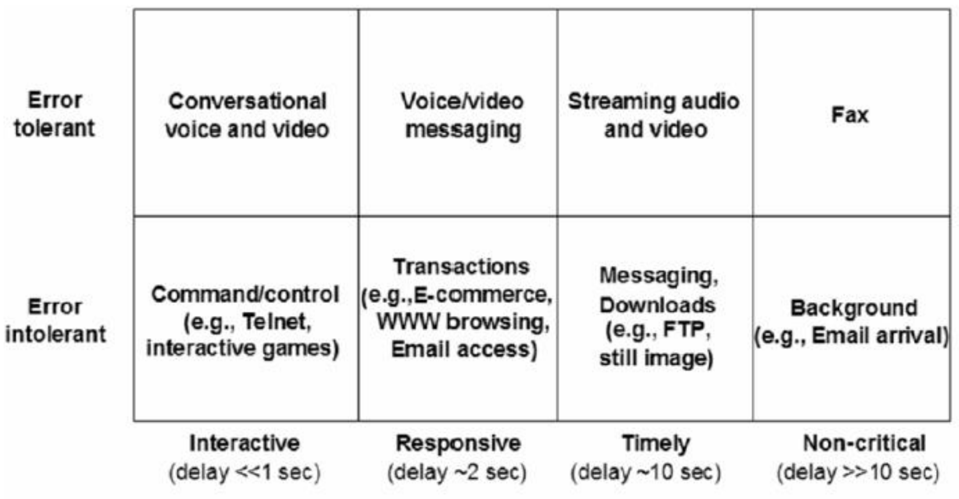
Jitter leads in a timing problem for the receiver. The jitter makes the decoding process in the receiver device complicated since the decoder fails to produce a smooth, continuous speech or continuous video stream. The receiving decompression algorithm requires fixed spacing data packets. The typical solution to jitter is to implement a de-jitter buffer within the receiver, so that packets are streamed with fixed spacing between them. The de-jitter buffer deliberately delays incoming packets in order to present them to the decompression algorithm at fixed spacing. The jitter buffer will also fix any out-of-order errors by looking at the sequence number in the RTP frames. The voice decompression engine receives packets directly on time, the individual packets are delayed further in transit, increasing the overall latency. Jitter causes either blocky, jerky or undesirable audio. Jitter for packets within a given stream should not exceed 20 - 50 milliseconds (Ngatman et al., 2008).

A lot of research on QoS has occurred, especially in wired networks. IntServ (Xiao & K.G. Seah, 2000)and DiffServ (Black et al., 1998) are two well-known, QoS models, designed for wired networks. Although much progress has been achieved on QoS for wire-based networks, a lot is still to be done when it comes to wireless networks. The unique characteristics like shared medium, mobility and the distributed multi-hop communication in wireless networks make it difficult to give a quality of service anticipated by the network user.

2.3.4 QoS Requirements for Voice, Video, and Data

The primary goal of QoS is to provide priority, including dedicated bandwidth, controlled jitter and latency (required by some real-time and interactive traffic), and improved loss characteristics. Moreover, it is important to assure that providing priority for one or more flows does not cause the failure of other flows. On intuitive level, QoS represents a certain type of requirements to be guaranteed to the users (e.g., how fast data can be transferred, how much the receiver has to wait, how correct the received data is likely to be, how much data is likely to be lost, etc.). Different standardization groups, like ITU, ETSI or

3GPP, have covered QoS requirements for multimedia traffic. They classified applications into eight groups, according to the error tolerance and delay, as summarized in Figure 3.1. QoS requirements and high-level recommendations for voice, video, and data are outlined in (Lewis & Pickavance., 2007) and (Rafael, Cacheda, Garc, & Gonz, 2007).



Error tolerant	Conversational voice and video	Voice/video messaging	Streaming audio and video	Fax
Error intolerant	Command/control (e.g., Telnet, interactive games)	Transactions (e.g.,E-commerce, WWW browsing, Email access)	Messaging, Downloads (e.g., FTP, still image)	Background (e.g., Email arrival)
	Interactive (delay <<1 sec)	Responsive (delay ~2 sec)	Timely (delay ~10 sec)	Non-critical (delay >>10 sec)

Figure 8: End-user QoS categories mapping. (Rafael et al., 2007)

Referring to Figure 8, it is possible to consider the following values on the ordinate axis for what concerns the error rates:

- Error tolerant applications
  - – Conversational voice/video *Frame Erasure Rate* (FER) < 3%
  - – Voice/video messaging FER < 3%
  - – Streaming audio/video FER < 1%
  - – Fax *Bit Error Rate* (BER) < 10^{-6}
- Error intolerant applications
  - – Information loss = 0.

Performance requirements for conversational services

The most common service in this category is real-time conversation, such as telephony speech. *Voice over IP* (VoIP) and video conferencing also belong to this category, with increasing relevance as the Internet is rapidly evolving. This is the only class whose characteristics are strictly determined by human perception (senses). Thus, this scheme has the most stringent QoS requirements: the transfer time should be low and, at the same time, the temporal relation of information entities of the stream should be preserved.

The limit for acceptable transfer delay is very strict (failure to provide low transfer delays will result in unacceptable lack of quality). However, there are loose requirements on FER, due to the human perception. For real-time conversation, the fundamental QoS characteristics are:

- Preserving the temporal relation of information entities in the same stream;
- Conversational pattern (stringent and low delay).

Some application examples based on conversational services are: conversational voice, videophone, interactive games, two-way control telemetry and Telnet.

Performance requirements for interactive services

This second class comprises interactive services (i.e., a human or a machine request on-line data from a remote server). It is characterized by the request response pattern of the end-user. An entity at the destination is usually expecting a response message within a certain period of time. The *Round Trip propagation Delay* (RTD) time is therefore one of the key attributes. Another characteristic is that the content of the packets must be transparently transferred (with a low BER). The resulting overall requirement for this communication scheme is to support interactive non-real-time services with low RTD.

For interactive traffic, the fundamental QoS characteristics are:

- The request-response pattern;
- Preserving payload content.

Some examples of this service type are: voice messaging and dictation, data, Web-browsing, high-priority transaction services (e-commerce) and e-mail (server access).

Performance requirements for streaming services

This service class is mainly unidirectional with high continuous utilization (few idle/silent periods) and low time variation between information entities within a flow. However, there is no strict limit for delay and delay variation, since the stream is normally aligned at the destination. Additionally, there is no strict upper limit for the packet loss rate.

For real-time streams, the fundamental QoS characteristics are:

- Unidirectional continuous stream;
- Preserving time relation (variation) between information entities of the stream.

The resulting overall requirement for this communication scheme is to support real-time streaming services with continuous unidirectional data flows. Table 1 (ITU-T, 2001) summarizes these applications providing the explicit requirements for each of them.

Table 1: Performance targets for audio, video and data applications

Medium	**Application**	**Typical data rates**	**Key performance parameters and target values**		
			One-way Delay	**Delay Variation**	**Information loss (Note 2)**
Audio	Conversational voice	4-64 kbit/s	<150 ms preferred (Note 1) <400 ms limit (Note 1)	< 1 ms	< 3% packet loss ratio (PLR)
Audio	Voice messaging	4-32 kbit/s	< 1 s for playback < 2 s for Record	< 1 ms	< 3% PLR
Audio	High quality streaming audio	16-128 kbit/s (Note 3)	< 10 s	<< 1 ms	< 1% PLR
Video	Videophone	16-384 kbit/s	< 150 ms preferred (Note 4) <400 ms limit		< 1% PLR
Video	One-way	16-384 kbit/s	< 10 s		< 1% PLR
Data Note 5	Web-browsing – HTML	~10 KB	Preferred < 2 s /page Acceptable < 4 s /page	N.A.	Zero
Data	Bulk data transfer/retrieval	10 KB-10 MB	Preferred < 15 s	N.A.	Zero

			Acceptable < 60 s		
Data	Transaction services – high priority e.g. e-commerce, ATM	< 10 KB	Preferred < 2 s Acceptable < 4 s	N.A.	Zero
Data	Command/control	~ 1 KB	< 250 ms	N.A.	Zero
Data	Still image	< 100 KB	Preferred < 15 s Acceptable < 60 s	N.A.	Zero
Data	Interactive games	< 1 KB	< 200 ms	N.A.	Zero
Data	Telnet	< 1 KB	< 200 ms	N.A.	Zero
Data	E-mail (server access)	< 10 KB	Preferred < 2 s Acceptable < 4 s	N.A.	Zero
Data	E-mail (server to server transfer)	< 10 KB	Can be several minutes	N.A.	$<10^{-6}$ BER
Data	Fax ("real-time")	< 10 KB	< 30 s/page	N.A.	$<10^{-6}$ BER
Data	Fax (store & forward)	< 10 KB	Can be several minutes	N.A.	Zero
Data	Low priority transactions	< 10 KB	< 30 s	N.A.	Zero
Data	Usenet	Can be 1 MB or more	Can be several minutes	N.A.	
NOTE 1 – Assumes adequate echo control. NOTE 2 – Exact values depend on specific codec, but assumes use of a packet loss concealment algorithm to minimize effect of packet loss. NOTE 3 – Quality is very dependent on codec type and bit-rate. NOTE 4 – These values are to be considered as long-term target values which may not be met by current technology. NOTE 5 – In some cases, for Data, it may be more appropriate to consider these values as response times.					

The choice of codec has impacts in many areas. The most important is the capacity planning on the network, because the bandwidth consumed in different codecs varies. When exploring the details of these needs in their work on tight IP service level agreement (SLA), G.114 states that 150 ms of end-to-end one-way delay does not cause a perceivable degradation in voice quality for most use of telephony. Some carriers try to push to the 100-ms target (excellent: 70 ms without propagation). A usual target is 150 ms (good: 120 ms without propagation).

It is also recommended that you look at the consumption of Layer 2 overhead; an accurate method for provisioning VoIP is to include the Layer 2 overhead. Layer 2 overhead includes preambles, headers, flags, cyclic redundancy checks (CRCs), and ATM cell padding. When Layer 2 overhead is included in the bandwidth calculations, the VoIP call bandwidth needs translate to the requirements shown in Table 2

Table 2: VoIP Bandwidth Reference Table

Codec	**Sampling Rate**	**Voice Payload in Bytes**	**Packets per Second (PPS)**	**Bandwidth per Conversation**
G.711	20ms	160	50	80 kbps
G.711	30ms	240	33	74 kbps
G.729A	20ms	20	50	24 kbps
G.729A	30ms	30	33	19 kbps

A more accurate method for the provisioning is to include the Layer 2 overhead in the bandwidth calculations, as shown in Table 3.

Table 3: VoIP Bandwidth Needs with Layer 2 Overhead

Codec	**801.Q Ethernet + 32 Layer 2 Bytes**	**MLP + 13 Layer 2 Bytes**	**Frame Relay + 8 Layer 2 Bytes**	**ATM + Variable Layer 2 Bytes (Cell Padding)**
G.711 at 50pps	93 kbps	86 kbps	84 kbps	104 kbps
G.711 at 33pps	83 kbps	78 kbps	77 kbps	84 kbps
G.711 at 50pps	37 kbps	30 kbps	28 kbps	43 kbps
G.711 at 33pps	27 kbps	22 kbps	21 kbps	28 kbps

Sample Calculation

We can use the following calculations to determine the inputs to the planning of voice call consumption:

$$Total\ Packet\ Size = (L2\ Header) + (IP, UDP, RTP\ header) + (Voice\ payload)$$

$$Parkets\ Per\ Second\ (pps) = \frac{Codec\ bit\ rate}{voice\ payload\ size}$$

$$Bandwidth = Total\ Packet\ size * Parkets\ Per\ Second$$

For example, the required bandwidth for a G.729 call (8-kbps codec bit rate) with cRTP, MP, and the default 20 bytes of voice payload is as follows:

$$Packet\ size = (MP\ header - 6bytes) + (Compressed\ IP, UDP, RTP\ header - 2bytes)$$
$$+(voice\ payload\ \ 20\ bytes = 28\ bytes = 224\ bits$$

$$pps = \frac{8kbps\ codec\ bit\ rate}{160\ bits} = 50\ pps$$

$$bandwidth\ per\ call = voice\ packet\ size\ (224\ bits) * 50\ pps = 11.2\ kbps$$

2.3.5 Human perception to QoS

Humans are much more sensitive to alterations of audio than visual signals. Our tolerance of transmission errors affecting audio streams is much lower than our tolerance of errors affecting motion video streams. If in application, audio and video are transmitted together, the two streams might compete for resources. In such cases the audio stream must have priority over the video stream. A good example is audio-video conferencing in packet mode supported by personal computers of workstations. The bit rate required for multimedia traffic depends on the quality and standard of technology used. Table 4 shows the required bit rate for various audio standards (Mahdi E. and Picovici D. 2006).

Table 4: Bit rates for audio streams

Quality	Technique/Standard	Bit Rate in Kbps
Telephone quality		
Standard	G.711 PCM	64
Standard	G.721ADCMP	32
Improved	G.722 SB-ADCMP	48, 56, 64
Lower	G728 LD-CELP	16
CD quality (stereo)		
Consumer CD audio	CD-DA	1411
Consumer CD audio	MPEG audio FFT	192
Improved(sound studio)	MPEG audio FFT	384

2.3.6 User Perceptive Quality of service

In the context of telecommunications, quality of service (QoS) definition borders on the degree of a user's satisfaction with the service. The QoS is thought to be divided into, speech or voice and video communication quality, service performance," and the necessary terminal equipment performance. The voice and video communication (or transmission) quality is more user-directed and, therefore, determines acceptability of the service from the user's point of view (Klaue, Rathke, & Wolisz, 2003). In this thesis we will call the voice and/or video transmission as multimedia transmission.

Although a lot of research has been devoted to mechanisms supporting the QoS in different types of networks, much less has been done to support the unified, comparable assessment of the quality really achieved by the individual approaches. Many researchers constrain themselves to proving that a certain mechanism is capable of reducing the packet loss rate, packet delay or packet jitter considering those measures as sufficient to characterize the quality of the resulting multimedia transmission. However, the above mentioned parameters cannot be easily and uniquely transformed into a quality of the transmission: in fact such transformation could be different for every coding scheme, loss concealment scheme and delay/jitter handling.

Quality can be defined as the result of the judgement of a perceived constitution of an entity with regard to its desired constitution. The perceived constitution contains the totality of the features of an entity. For the perceiving person it is a characteristic of the identity of the entity. Applying this definition to multimedia, voice and video quality can be regarded as the result of a perception and assessment process, during which the assessing subject establishes a relationship between the perceived and the desired or expected multimedia signal. In other words, multimedia quality can be defined as the result of the subject's judgement on spoken language, which he/she perceives in a specific situation and judges instantaneously according to his/her experience, motivation, and expectation. Regarding voice communication systems, quality is the customer's perception of a service or product, and multimedia quality measurement is a means of measuring customer experience of telecommunication services. The most accurate method of measuring multimedia quality therefore would be to actually ask the callers during or after the call, for their opinion on the quality (Klaue et al., 2003).

Table 5: Listening-quality scale

Score	**Quality of speech**	**Impairment**
5	Excellent	Imperceptible
4	Good	Perceptible, but not annoying
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Very annoying

In practice, there are two broad classes of voice quality metrics: subjective and objective. Objective measurements uses instruments to measure quality of service metrics like delay, packet loss and jitter. This type of measurement is easy to work with or to compare the performance of two systems since it uses measured units. Subjective measurements, known as subjective tests, are conducted by using a panel of people to assess the voice quality of live or recorded speech signals from the voice communication system/device under test for various adverse distortion conditions. Here, the speech quality is expressed in terms of various forms of a mean opinion score (MOS), which is the average quality perceived by the members of the panel as shown in Table 5.

2.4 Quality of Service Issues in MANETs

While it is difficult to provide quality of service in wired networks, MANETs and wireless networks in general bring in more difficulties because of their characteristics. The nature of wireless networks of being error-prone and high mobility makes it difficult to integrate traditional Internet QoS protocols to wireless networks.

The main objective of QoS in MANETs is to achieve a more deterministic network behaviour so that information carried by the network can be better delivered and network resources are best utilized. This can be achieved by raising the priority of a traffic flow or limiting the priority of another flow.
Since multimedia requires high bandwidth, getting it to work over fixed connections is hard enough, but now we require it to work efficiently on MANETs also. Besides the known interference problems faced by wireless networks, MANETs have their own characteristics that bring challenges in guaranteeing quality of service (Mohapatra et al., 2003). These include the following:

- Node mobility and non-infrastructure: Node mobility is the basic cause of the dynamic network topologies in MANETs. When nodes move, the MAC layer allocation of bandwidth to each node also changes. Bandwidth is difficult to control due to the non-infrastructure feature coupled with the continual changing of roles for nodes from router to host and the dynamic topology of the MANET. In MANETs there is no central infrastructure that can regulate the distribution of resources to nodes. The network is decentralized, where all network activity including discovering the topology and delivering messages must be executed by the nodes themselves, that is, routing functionality will be incorporated into mobile nodes. The nodes are free to move about and organize themselves into a network, thus the network topology may change rapidly and unpredictably over time. The challenge here is to design a decentralized QoS schemes.

- Limited Bandwidth and Network Size: It would appear as if scalability is an unlikely problem for MANETs since they are mostly of small size. However as technology advances rapidly the emergence of high-speed and large-sized MANETs with plenty of applications is foreseeable in the near future, thus bringing with it the scalability problem.
- Time-Varying Feature: Link capacity in the wireless environment is time dependent due to factors such as fading and shadowing, the dynamics of the network topology and mobility of nodes. This feature makes the QoS provision in MANETs a very difficult task. Take the signalling protocol for example. A signalling protocol generally comprises three phases: connection establishment, connection teardown, and connection maintenance. It is predicted that a large proportion of link capacity will be occupied by control overhead in a MANETs. For MANETs the overheads of connection maintenance may actually outweigh the initial cost of establishing the connection (Xiao & K.G. Seah, 2000).
- Power Constraints: The nodes' processing capability is limited due to the limited battery power. This means there should be low processing overheads of nodes and thus, the control algorithms and QoS algorithms should use bandwidth and energy efficiently. QoS challenges due to limited capabilities of mobile nodes in terms of processing power, storage capacity, or energy. The limited capabilities challenge, influence, and shape the QoS design for instance by forcing a distributed approach, avoiding lookup tables, accommodating dormant devices, or adopting simpler lightweight algorithms.
- QoS challenges due to the lack of central authority that can maintain central information on flows, routes, or connections.
- QoS challenges due to Hidden and Exposed Terminal Problems: In a MAC layer with the traditional carrier sense multiple access (CSMA) protocol, multi-hop packet relaying introduces the "hidden terminal" and "exposed

terminal” problems. The hidden terminal problem happens when signals of two nodes, say A and C, that are out of each other’s transmission ranges collide at a common receiver, say node B (see Figure 9) (Jayasuriya et al., 2005).

- An exposed terminal problem will result from a scenario where node B attempts to transmit data A while node C is transmitting to node D. In such a case, node B is exposed to the transmission range of node C and thus defers its transmission even though it would not interfere with the reception at node D (see Figure 10) (Jayasuriya et al., 2005).

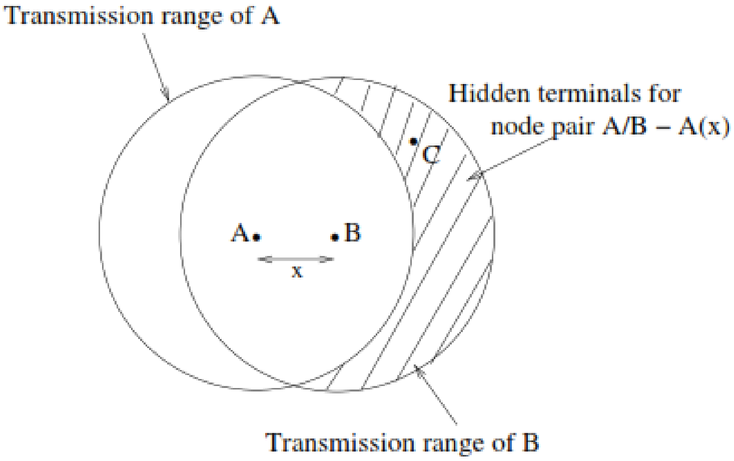


Figure 9: An illustration of Hidden terminal problem

All these challenges lead to serious concern in the provision of quality of service in ad-hoc networks. Some of these challenges influence greatly the issue of flow reservation in ad-hoc networks.

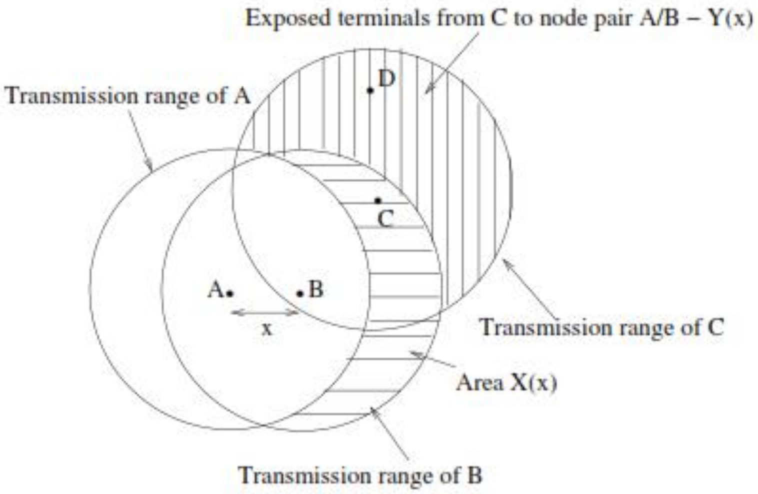


Figure 10: An illustration of Exposed Node Problem

2.5 Wireless Sensor Networks

Recent technological advances, like the availability of low-cost hardware such as CMOS cameras and microphones, have enabled the development of low-cost, low-power, and multi-functional sensor devices with integrated sensing, processing, and communication capabilities. A sensor is an electronic device that is capable of detecting environmental conditions such as temperature, sound, chemicals, or the presence of certain objects. The sensing circuitry measures parameters from the environment surrounding the sensor and transforms them into electric signals.

Figure 11 illustrates a typical sensor network used in a national park to curb fires. Processing such signals reveals some properties of objects located and/or events happening in the vicinity of the sensor. When a fire is detected an alarm message (arrow) is generated by the sensor node(s) that detected the fire and relayed through the network until it reaches a park ranger.

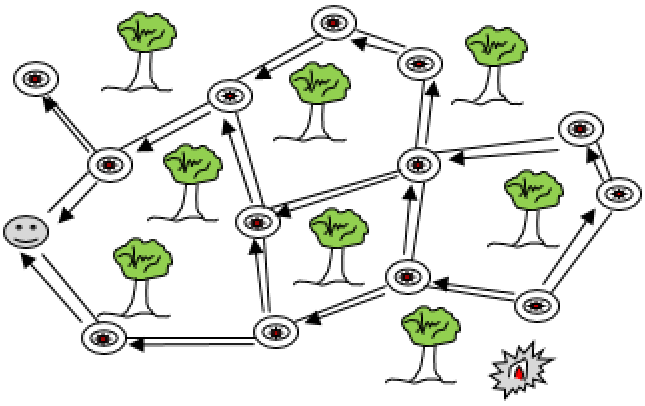


Figure 11: A sensor network used for prompt fire detection

Wireless sensor devices can be networked together in a number of ways to implement specific applications. In basic data-gathering applications, for instance, there is a node referred to as the sink to which all data from source sensor nodes are directed. In some cases all nodes would send their data directly to the sink using single hop topology. In networks deployed over a large area, a multi-hop tree structure may be used for data-gathering and nodes act both as sources themselves, as well as routers for other sources, thereby forming a wireless ad hoc network that link to the sink. Wireless sensor networks are expected to be applied in ecological habitat monitoring, structured health monitoring, environmental contaminant detection, industrial process control, and military target tracking. Many other applications that can take advantage of the capabilities of wireless ad hoc networking will arise from time to time.

One interesting characteristic of wireless sensor networks is that they often allow for the possibility of intelligent in-network processing. In this case intermediate nodes may also examine and process the content of the forwarded packets. This makes the wireless sensor network more useful and improves the quality of collected data. Data gathering in sensor networks differs from the general ad hoc network's data communication protocols. Sensors in general monitor or measure the same event or data and report it to the sink. The data from many sensors may be combined en-route to the sink, to save energy and increase reliability of reports. In some cases, sensor data indicate detection of a target, while fusion of multiple sensor reports can be used for tracking and identifying the detected target. These are sometimes called reconnaissance-oriented networks. When sensors are connected to the Internet, it brings in the concept of the Internet of things. From this point on, this thesis will assume that everything that applies to MANETs also applies to wireless sensor networks.

2.6 Smart phone Mobile ad Hoc networks

While smartphones are now ubiquitous, most mobile applications still use a client-server model rather than communicating as a MANET. Smartphones are a potentially useful tool when there is no network infrastructure, such as in disaster recovery situations or for field soldiers in a combat environment. They may also be useful when communication is between nearby devices, thereby avoiding mobile operator data charges, or evading administrative control during civil demonstrations and strikes. Currently, smartphones operate using network operators' base stations. However, for emergencies and other situations, they must be able to operate independently of a commercial cell-phone network as a MANET.

In recent years, researchers have been coming up with MANET and routing protocol implementations on the smartphone. The operating system of a smartphone provides different levels of capability and different permissions to users and developers with respect to MANET development and use. As a baseline, no smartphone provides the ability to operate as a backbone node within a MANET. However, some operating systems do allow rooting or jailbreaking to allow a user of the smartphone to operate in privileged mode and to develop and install software that allows routing packets from one phone to the next. In this section, we look at some of various developments that emerged over the years in creating MANETs using smartphones.

2.6.1 WiFi Direct

WiFi Direct is a technical specification (Wi-Fi-Alliance, 2010) of the WiFi Alliance that leverages existing standards to provide a convenient way for securely connecting devices without installed infrastructure, enhanced with features like peer and service discovery. It is based on the independent basic service set (IBSS) mode of IEEE 802.11 in such a way that one of the devices is to be the group owner (Group Owner (GO)), through negotiation, and acts as an Access Point (AP). All other devices will connect to the network through the GO forming a star topology. The GO incorporates a Dynamic Host Configuration Protocol (DHCP) server for providing IP addresses to the client nodes.

While the specification mentions Concurrent Devices that can simultaneously connect to the infrastructure or be part of a different group, additional protocols are required for routing. A significant disadvantage of WiFi Direct is that if the GO leaves, the group is torn down and a new group must be established from scratch. While these limitations are irrelevant in simple situations like a printer letting computers and other devices connect, they make WiFi Direct unsuitable as a basis for multi-hop networking. WiFi Direct in Android assigns the same IP address (192.168.49.1) to the GO of all groups making the resultant network vulnerable to security breaches. These issues reveal the inadequacy of using WiFi Direct for multi-hop networks, particularly in Android.

2.6.2 Melon

MELON (Collins & Bagrodia, 2014) is a general-purpose coordination language designed to provide flexible communication patterns for MANET applications while remaining lightweight. Based on a distributed shared message store, MELON abstracts network communication to an asynchronous exchange of persistent messages. MELON simplifies application development by supporting read-only and remove-only messages, bulk message retrieval, and per-host ordering of messages.

The design of MELON centres on a distributed shared message store. Each device in the network may host any number of applications, which access and contribute to the shared message store. Each application hosts a local message store, which may be accessed by any other local or remote application. Messages are sent and received asynchronously by storing and retrieving them from the shared message store, removing the need for a persistent connection.

This provides temporal decoupling between hosts, since messages can still be delivered even after prolonged disconnections. Discovery of available messages is performed on demand for each operation.

While this does increase the amount of communication required for each operation, it avoids global state and allows the network to change at any time. MELON also provides spatial decoupling by matching messages based on content, instead of a host address or location. The messages themselves may physically reside on any host in the network. The sender of a message is not aware of the receivers' identities nor even how many receivers might read a message. This frees applications from tracking remote addresses or contacting a directory service to find remote resources. MELON supports multicast communication by allowing any number of receivers to read the same message. MELON also provides bulk receives, which allow applications to efficiently receive multiple messages from multiple hosts in a single operation. Applications often also require unicast communication. While unicast communication can be accomplished by storing regular messages in MELON, these can be disrupted by a process removing a message intended for a different receiver. It is also possible to eavesdrop on messages by reading but not removing a message. For applications such as instant messaging, it is important to have private unicast communication.

2.6.3 Open Garden

Open Garden (Iosifidis, Gao, Huang, & Tassiulas, 2014) is a software for Internet connection sharing on mobile devices using a mesh of Bluetooth or WiFi Direct links. It also allows communication between devices across multiple hops as long as the application uses Open Garden’s proprietary forwarding software. The FireChat application, from the same company, runs on top of Open Garden enabling a multihop messaging framework. Open Garden works by creating a Virtual Private Network (VPN) to a Bluetooth paired device also running the application. The other device terminates the VPN tunnel and either forwards the request to another node or redirects the message to the local application that registered for it (most commonly FireChat). With this architecture, a multi-hop overlay network is established using Bluetooth connections. From the analysis done by (Soares, Brandão, Prior, & Aguiar, 2017) they concluded that no IP level connectivity that might be used by other applications is provided in Open Garden.

2.6.4 Serval Project

The Serval project (Gardner-Stephen & Palaniswamy, 2011) provides a free and open-source software to allow mobile phones to communicate in the absence of phone towers and other infrastructure, targeting disaster situations and remote communities. The Serval Mesh application provides voice calls, text messaging and file sharing directly over IEEE 802.11 links between mobile devices. It can be used for peer-to-peer communication through an IEEE 802.11 API or in an ad-hoc multi-hop topology without infrastructure support. The MANET is implemented using an ad-hoc routing protocol over IEEE 802.11 in IBSS mode. The project initially used BATMAN, but moved to an in-house routing protocol. The project developed Mesh Datagram Protocol (MDP), a hybrid of network and transport layer protocol that shares some properties with User Datagram Protocol (UDP), but with per-hop retransmission of packets for mitigating the cumulative end-to-end packet loss effect that can significantly affect the performance of multi-hop wireless environments. MDP can work over IP, or directly over link layer technologies. On top of MDP, the project provides Rhizome, a resilient file distribution protocol that is used to transparently transport data across the mesh nodes. It is used for transmitting messages or support other services, such as their Voice over Mesh Protocol (VoMP). The project also defines a Distributed Numbering Architecture (DNA) to identify and address the nodes with cryptographic IDs on the network.

The current application on the Google Play Store includes the Serval Mesh that provides the above functionality including the project's routing protocol. Currently the development is being driven for mobile phones and Android is the one currently supported with applications. The specificities of the protocols outlined above make the Serval approach unusable by applications that are not aware of their API and sub-system. This provides little to no flexibility as a MANET test-bed.

2.6.5 SPAN

Motivated by the crash of the cell phone network in Haiti after the devastating earthquake in 2010, Josh Thomas and Jeff Robble, decided to create a working prototype MANET using only the Wi-Fi chips on Android smartphones (Thomas & Robble, 2012). This Smart Phone Ad-Hoc Networks (SPAN) project reconfigures the on-board Wi-Fi chip of a smartphone to act as a Wi-Fi router with other nearby similarly configured smartphones, creating a MANET without an operational carrier network. SPAN intercepts all communications at the Global Handset Proxy so applications such as VoIP, Twitter, email etc., work normally.

They merged source from the Linux Wireless Extension API into the Android kernel source and compiled it. They used this modified version of Android to root Android smartphones to expose and harness the ad-hoc routing features of the on-board Wi-Fi chip to enable this intercept. The researchers designed SPAN in such a way that its routing protocol is plug-and-play and so that can be easily replaced. SPANs differ from traditional hub and spoke networks, such as Wi-Fi Direct, in that they support multi-hop relays and there is no notion of a group leader so peers can join and leave at will without destroying the network. SPAN is a framework for further research to refine how to build the special case of an ad-hoc mesh network. Span is still work in progress that will influence the way mobile devices will work in the future.

2.6.6 AdHocDroid

Researchers in (Soares et al., 2017) developed an IP-based mesh network, they called AdHocDroid. This network provides a framework which supports standard TCP/IP applications and generates a genuine IEEE 802.11 MANET, initially using Android-based smartphones which have been rooted. AdHocDroid first systematically disables network connectivity using the Android API, in order to stop individual applications attempting to re-enable or reconfigure network settings on the phone. The application then rewrites the text file which stores all known network configuration in order to make the MANET the default connectivity path.

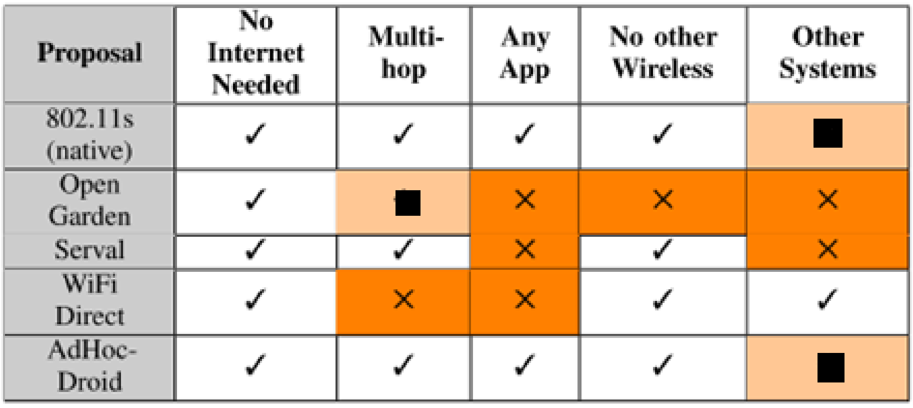
2.6.7 Comparison of smart phone MANET technologies

Given that some applications claim to provide mobile ad-hoc networking, (Soares et al., 2017) set some basic requirements of what defines a MANET and they concluded that a systems should satisfy the following requirements:

- Is communication possible without connectivity to the Internet? (No Internet Needed)
- Is multi-hop communication possible? (Multi-hop)
- Can any application take advantage of the provided connectivity through a regular socket API, thus not requiring adaptation/re-writing? (Any App)
- Can work without needing additional wireless technology to provide communication, e.g.: using IEEE 802.11 needs also BT? (No other Wireless)
- Can we use off-the-shelf OS's to communicate with the MANET? E.g.: if development is on Android can we communicate with a PC running another OS? (Other Systems)

They analysed these technologies according to this definition, and summarised the results in Table 6.

Table 6: Manets network solutions checklist ((Soares et al., 2017)



Proposal	No Internet Needed	Multi-hop	Any App	No other Wireless	Other Systems
802.11s (native)	✓	✓	✓	✓	■
Open Garden	✓	■	×	×	×
Serval	✓	✓	×	✓	×
WiFi Direct	✓	×	×	✓	✓
AdHoc-Droid	✓	✓	✓	✓	■

A yes is denoted by a √, a no is denoted by × and ▪ indicates partially OR with some adaptations. As is summarized in the table, only AdHocDroid and 802.11s truly provide all the features for a MANET. WiFi Direct cannot perform multi-hop, in addition to requiring a dedicated framework; Open Garden shares this latter flaw, in addition to requiring a wireless network at one point in the mesh; while Serval again needs dedicated systems. However, WiFi Direct is currently the only MANET which has support for other systems, such as the PC platform – functionality that's partly enabled on AdHocDroid and completely absent from Open Garden and Serval.

2.7 Quality of Service Models for MANETs

2.7.1 Introduction

A QoS model specifies the architecture in which a network can provide certain services whilst considering the challenges of MANETs. These challenges include dynamic topology, low bandwidth and time-varying link capacity. MANETs would require a seamless connection to the Internet in order for them to be used in commercial applications. Therefore, the QoS model should also consider other existing QoS architectures in the Internet. In this section, we discuss existing QoS models for the Internet and MANETs, IntServ and DiffServ, INSIGNIA, SWAN and ASAP.

2.7.2 IntServ

The IntServ model (Braden et al., 1994) merges the advantages of datagram networks and circuit switched networks. It can provide a circuit-switched service in packet-switched networks. In circuit-switching, this path is decided upon and established before the data transmission starts. For the whole communication session, the route is dedicated and exclusive, and released only when the session terminates. In packet-switching there is no predetermined path and packets are sent towards the destination independent of each other. Each packet finds its own path to the destination making routing decisions at various nodes in the path.

The Resource Reservation Protocol (RSVP) was designed as the primary signalling protocol to setup and maintain the virtual connection. RSVP is also used to propagate the attributes of the data flow and to request resources along the path. Routers finally apply corresponding resource management schemes to support QoS specifications of the connection. Based on these mechanisms, IntServ provides quantitative QoS for every flow.

Resource Reservation Protocol (RSVP)

The (RSVP) (Wroclawski, 1997) is a classic two-pass protocol using out-of-band signalling. Figure 2.7 shows the classical operation of RSVP. The messages used are the Path message, which originates from the traffic sender and the reservation (Resv) message, which originates from the traffic receivers. The primary roles of the *Path* message are first to install reverse routing state in each router along the path, and second to provide receivers with information about the characteristics of the sender traffic and end-to-end path so that they can make appropriate reservation requests. *Resv* messages finally carry reservation requests to the routers along the distribution tree between receivers and senders. RSVP state is "soft-state", after a certain expire time, the state of the path and the reserved resource is released. Periodical issuing of *Path* or *Resv* messages are necessary to keep the reservation alive. Additional signalling information allows the soft state timeout to adapt to the refresh period. Furthermore, RSVP provides a routing triggered local repair [8] mechanism to overcome the need for a very fast refresh rate in order to react to route changes.

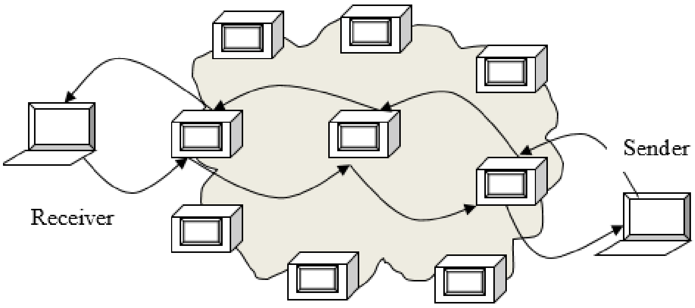


Figure 12: The operation of RSVP

Disadvantages of IntServ/RSVP

The short comings of IntServ in MANET environments are in scalability and signalling. The amount of state information in IntServ increases proportionally with the number of flows since IntServ provides per flow granularity. Keeping flow state information will cost a huge storage and processing overhead for the mobile host whose storage and computing capacity are scarce. The scalability problem is less likely to occur in current MANETs considering the small number of flows, the limited size of the network and the bandwidth of the wireless links. However, as the quality of wireless technology increases rapidly, high speed and large size MANETs may be found in the future and the problem will manifest.

The signalling protocols have three phases: connection establishment, connection maintenance and connection teardown. Since MANETs have very dynamic topologies, this approach is not reliable since routes may change very quickly and the handshaking would not be fast enough. Due to its out-of-band approach, RSVP produces a significant signalling overhead. This means that RSVP signalling packets will contend for bandwidth with data packets and consume a substantial amount of bandwidth in MANETs. This may be of importance if the refresh rate is high because the message size is not negligible in RSVP. A high refresh rate might occur when no route-change notification service from the routing layer is available. This causes local repair to fail.

2.7.3 Differentiated Service (DiffServ)

DiffServ (Black et al., 1998), was designed to overcome the difficulty of implementing and deploying IntServ and RSVP in the computer network (K. Wu and J. Harms 2001). IntServ provides per-flow guarantees but Differentiated Services (DiffServ) maps multiple flows into a few service levels. DiffServ defines three types of nodes. An ingress node is a source node and an egress node is a destination node. Intermediate nodes are called Interior nodes and they are responsible for forwarding data on behalf of other nodes. At the boundary of the network, traffic entering a network is classified, conditioned and assigned to different behaviour aggregates by marking a special DS (Differentiated Services) field in the IP packet header which supersedes the TOS field in IPv4 and CLASS field in IPv6. Within the core of the network, interior nodes, packets are forwarded according to the per-hop behaviour (PHB) associated with the DSCP (Differentiated Service Code Point). An intermediate network node performs a PHB, which is a logical instantiation performing traffic forward behaviour. The forward behaviour normally follows the traffic resource allocation per link based on the priority defined in DSCP. The traffic resource is determined based on packet loss rate, propagation delay and jitter. This eliminates the need to keep any flow state information elsewhere in the network (K. Wu and J. Harms 2001).

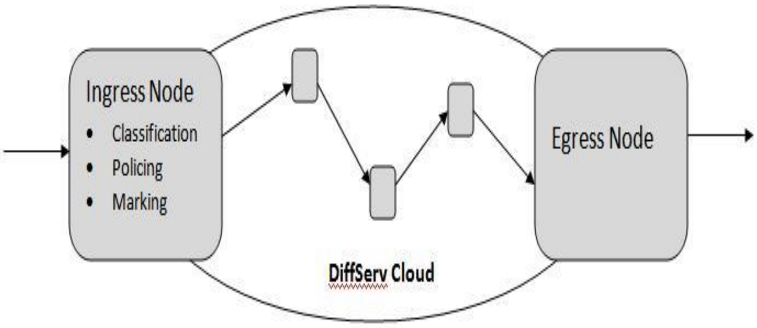


Figure 13: The DiffServ Architecture (Iqbal, 2008)

The main drawbacks of a DiffServ approach in MANETs are listed below:

- Soft QoS guarantees: DiffServ uses a relative-priority scheme to map the quality of service requirements to a service level. This aggregation results in a more scalable but also in more approximate service to user flow.
- SLA (Service Level Agreement): DiffServ is based on the concept of SLA's which are contracts between customers and their Internet Service Providers (ISPs) that specify the forwarding service each customer should receive. In a DiffServ domain it is important that sufficient resources are provisioned to support the SLA's committed by the domain. Also, the boundary nodes must monitor the arriving traffic for each service class and they should perform traffic classification and conditioning to enforce the negotiated SLA's. If a customer acquires QoS parameters and he pays for such parameters then there must be some entity which will assure them. In a completely ad hoc topology where

there is no concept of service provider and client and where there are only clients it would be quite difficult to innovate QoS, since there is no obligation from somebody to somebody else what makes QoS almost infeasible.

- Ambiguous core network: The benefit of DiffServ is that traffic classification and conditioning only has to be done at the boundary nodes. This makes quality of service provisioning much easier in the core of the network. In MANETs though there is no clear definition of what is the core network because every node is a potential sender, receiver and router. This means that several separate flow states will be maintained at intermediate nodes just like in IntServ (K. Wu and J. Harms 2001).

2.7.4 FQMM

Flexible Quality of Service Model for Mobile Ad Hoc Networks (FQMM) (Xiao & K.G. Seah, 2000), is another QoS model which was designed to combine the IntServ and the DiffServ models in order to combine the strengths of the two models whist at the same time trying to override the weaknesses. Ingress, Engress and Interior nodes are defined, exactly the same way as in DiffServ.

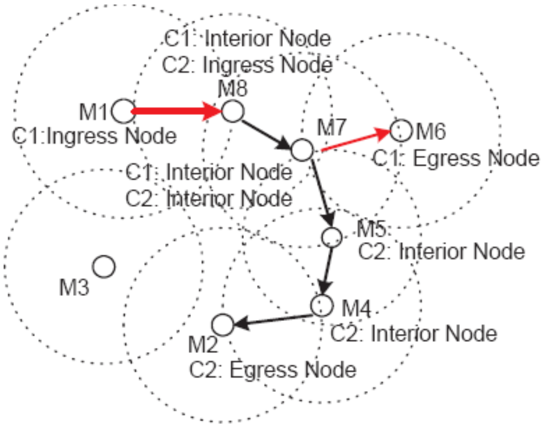


Figure 14: An illustration of type of nodes in FQMM depending on scenarios.

Figure 14 shows a scenario where there are two connections: one is from M1 to M6 and another from M8 to M2. The roles of the nodes change depending on what part they are playing for a specific flow. Node M8 is an interior node for flow C1 and it is an Ingress node for flow C2.

FQMM combines both the service differentiation of DiffServ and the per-flow state property of IntServ by both preserving per-flow granularity for a small portion of traffic in the MANET, and per-class granularity. A traffic conditioner is placed at the ingress nodes where the traffic originates. Components of the conditioner include traffic profile, meter, marker and dropper. The traffic profile decides the policy of other components which change the configuration according to the traffic profile. It is responsible for re-marking or discarding packets according to the traffic profile, which describes the temporal properties of the traffic stream such as transmission rate and burst size.

FQMM is a first and important attempt at proposing a QoS model for MANETs. It however suffers of major problems:

- FQMM aims to tackle the scalability problem of IntServ. However there is no explicit control on the number of services with per-flow granularity, therefore the problem still exists (Khalid Iqbal, 2005).
- Due to its DiffServ behaviour in ingress nodes, FQMM may not be able to satisfy hard QoS requirements. It could be difficult to code the PHB in the DS field if the PHB includes per-flow granularity, considering the DS field is at most 8 bits without extension (K. Wu and J. Harms, 2001).
- How to make a dynamically negotiated traffic profile is a well-known DiffServ problem and FQMM seems not to solve it (Parvez & Peer, 2010)

2.7.5 INSIGNIA

INSIGNIA (S.-B. Lee, Ahn, Zhang, & Campbell, 2000) is a signalling protocol designed explicitly for MANETs. It can be combined with a variety of routing protocols to come up with an effective QoS model. It supports fast flow reservation, restoration and adaptation algorithms that are specifically designed to deliver adaptive real-time service. INSIGNIA implements an in-band signalling approach by encapsulating some control signals in the IP option of every data packet, which is now called INSIGNIA option, Figure 15.

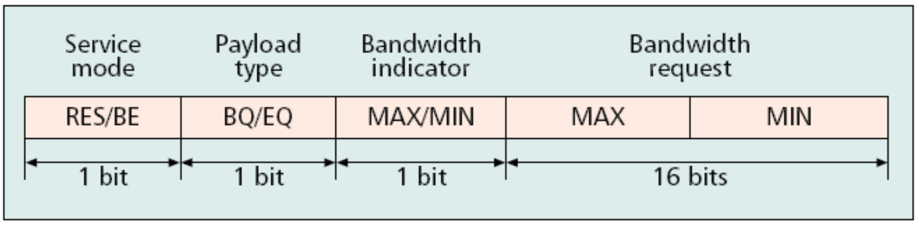


Figure 15: The INSIGNIA IP option in a packet.

Flow state information is kept in every node in a particular path. This is done in such a way that, the flow state information is periodically refreshed by the received signalling information. This is called soft-state reservation. When a source node wants to establish a reservation to a destination node it sets the *reservation* (RES) mode bit in the INSIGNIA IP option service mode of a data packet and sends the packet towards the destination. The bandwidth request field allows a source to specify its maximum (MAX) and minimum (MIN) bandwidth requirements. Intermediate nodes execute admission control upon receiving a RES bit. When the node accepts a request, resources are committed for the particular flow and subsequent packets are scheduled accordingly. When the request is denied there is no reservation and packets are treated as *best effort (BE)* mode packets.

In the case where a RES packet is received and no resources have been allocated, the admission controller attempts to make a new reservation. This is a re-active local repair mechanism and commonly occurs when flows are rerouted during the lifetime of an ongoing session due to host mobility. When a node receives a request packet with the bandwidth indicator bit set to MAX indicates that all nodes before this node have enough resources to support the maximum bandwidth requested. If the bandwidth indicator is set to MIN it implies that at least one of the intermediate nodes is a bottleneck node and the maximum bandwidth requirement may not be met. As a result "partial reservations" will exist between source and bottleneck node, these resources remain reserved until explicitly released.

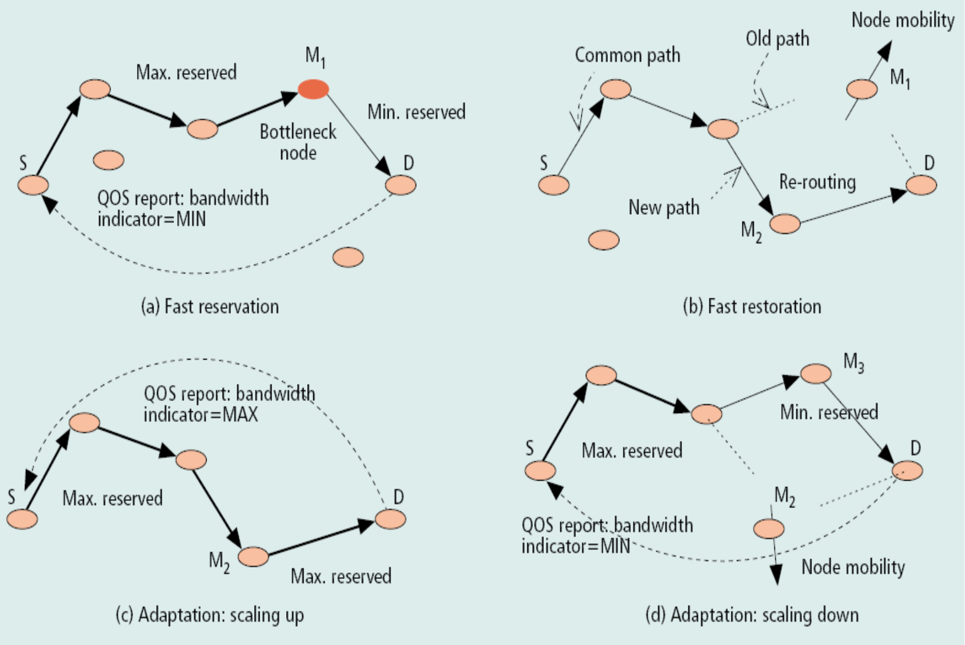


Figure 16: Examples of INSIGNIA operations.(Stuedi, Xue, & Alonso, 2004)

When a reservation is received at the destination node, INSIGNIA checks the reservation establishment status. QoS reporting message can be sent by destination nodes to inform source nodes of the ongoing status of flows. These messages do not have to travel on the reverse path toward a node. The report commands can either be scale-down or scale-up commands. A scale-down command requests a source either to send with the rate specified as MINIMUM instead of MAXIMUM or to send its packets as best effort instead of MINIMUM depending on the current sending rate of the source node. This will clear any partial reservations. A scale up requests a source node to initiate a reservation for some MINIMUM or MAXIMUM rate, depending on the actual flow state. Figure 16 shows various INSIGNIA options depending on the condition of the network.

Disadvantages of INSIGNIA in MANETs

The most obvious drawback of INSIGNIA is its scalability problem due to the flow state information which is kept within the nodes of a certain path.

INSIGNIA’s bandwidth usage is not efficient. The extra reservation on the path from the sending node to the bottleneck is a waste of bandwidth until an explicit release message is sent. Although this waste won’t last long, topology changing of MANET will make this reservation waste propagate frequently. Furthermore releasing partial reservations using QoS reports enforces source nodes either to set the bandwidth indicator of the INSIGNIA option field to MINIMUM or to send the packets as best effort depending on the actual flow state. In both cases the opportunity to scale up is lost.

INSIGNIA does not provide any mechanism to dynamically change the frequency by which control signals are inserted into the data packets. This imposes a major processing overhead on the network. Only two bandwidth levels to be used are offered, MINIMUM and MAXIMUM. A more fine-grained approach would be needed in order to satisfy application requirements and to fully exploit the resources available. INSIGNIA differentiates traffic into best effort (BE) and Quality of Service traffic which was split into base or enhanced quality of service (BE/EQ) depending on the payload of the network. Multimedia traffic however, comes in different types varying from online games, internet telephone, video conferencing, video streaming and many others. These different kinds of traffic have to be treated with different priorities by the network just like it is done in DiffServ.

In INSIGNIA if the available bandwidth is just enough to only meet the minimum bandwidth requirement needs of the base QOS, enhanced QOS packets are degraded to best effort packets at bottleneck nodes by changing the service mode for EQ packets from RES to BE. When a node encounters degraded packets, it releases bandwidth that would have been allocated to enhanced QOS packets. Whilst this releases unused resources in intermediate nodes, it does not give a guarantee that the flow received will be at a useful quality of service level since it is below the minimum required level. There is a need to keep alive only those flows that have guaranteed quality of service level.

2.7.6 SWAN

Stateless Wireless Ad-Hoc Networks (SWAN), (Ahn, Campbell, Veres, & Sun, 2002) is a stateless network model designed to provide service differentiation in MANETs that employ a best-effort distributed MAC. It classifies traffic into real-time UDP traffic and best-effort UDP and TCP traffic using a classifier. A leaky-bucket traffic shaper that applies the Additive Increase Multiplicative Decrease (AIMD) rate control algorithm controls the rate of best-effort packets. The AIMD rate controller uses per hop delay, measured at each node and restricts the bandwidth for best-effort traffic in favour of real-time applications. The bandwidth not used by real-time applications can then be allocated to best-effort traffic. To avoid excessive delays, the total traffic, both best-effort and real-time, transported over wireless channel is maintained below a certain threshold value.

SWAN uses sender-based admission control for real-time traffic. Each node measures the rate of aggregated traffic and the source node uses this as feedback coupled with an end-to-end probe to estimate the local bandwidth availability. The source node dispatches a UDP probe packet, which has to be processed by all intermediate nodes, towards the destination. This probe packet contains a "bottleneck bandwidth" field which all intermediate nodes compare to their available bandwidth. If the available bandwidth of a node is less than the bottleneck bandwidth it updates the bottleneck field with its own bandwidth. The available bandwidth is the difference between an admission threshold value and the current rate of real-time traffic. When the destination node receives the probe packet it returns a response packet with the bottleneck bandwidth back to the source. Upon receiving the response the source node compares the end-to-end bottleneck bandwidth and the bandwidth requirement and then decides whether or not to admit a real-time flow. The admitted real-time packets are not regulated at the intermediate node. They are marked as RT (real-time packets) and they bypass the shaper mechanism at the intermediate nodes.

SWAN offers soft QoS. This means that, when traffic load conditions and network topology change, real-time sessions might fail to get the minimum bandwidth and delay requirements and they must be rejected or readmitted. No flow information is kept in the intermediate nodes to avoid complex signalling and state control mechanisms (Ahn et al., 2002).

Disadvantages of SWAN

It is unclear how the amount of bandwidth available for RT traffic should be chosen in a sensible way. Choosing larger value results in a poor performance of RT flows and starvation of BE flows, and choosing too low values results in the denial of RT flows for which the available resource would have sufficed. There would also be no flexibility to tolerate channel dynamics. The total rate of aggregated RT traffic may be dynamic due to node changes in traffic patterns and node mobility. Due to node mobility, for example, intermediate nodes may need to maintain RT traffic in excess of resources set-a-side for RT traffic. An intermediate router making this observation sets the explicit congestion notification flag in RT packets' headers.

In addition, source-based admission control using probing packets is again unrealistic and ineffective in a dynamic environment such as a MANET, as conditions and network topology tend to change fairly frequently. Furthermore, bandwidth calculations in SWAN do not take best effort traffic into consideration, and hence may lead to a false estimation of the available bandwidth (Parvez & Peer, 2010). The other limitation of SWAN is that it does not require the support of a QoS capable MAC. Instead, soft real-time services are built using existing best effort wireless MAC technology. Thus, though SWAN can be a candidate QoS model, it cannot be a complete QoS solution for a highly dynamic network like MANETs. We can conclude that SWAN tries to maintain delay and bandwidth requirements of RT traffic by admission control of UDP traffic and rate control of TCP and UDP traffic. SWAN fails to fully utilise the DiffServ field which they use only for two classes of traffic. It would be more useful if they have taken full advantage of differentiating the traffic into various classes that are out there in practice.

2.7.7 ASAP

Adaptive Reservation and Pre-Allocation Protocol (ASAP), (Xue, Studi, & Alonso, 2003) provides adaptive QoS support to real time applications in infrastructure based wireless IP networks. The purpose of this analysis is to extend the ASAP framework which can be used in mobile ad hoc networks. In ASAP architecture, a reservation concept, soft/hard reservation is introduced for efficient resource allocation as shown in Figure 17 and Figure 18. Soft reservation can be considered as the claim of a traffic flow for a certain bandwidth to be used in future. Hard reservation enables a traffic flow to exclusively reserve some bandwidth.

The actual reservation mechanism is two pass based. When a new real-time flow is about to start, a soft reservation request is sent first. If there are enough resources available, the requested bandwidth will be soft reserved for that flow. After a soft reservation is established, the end node sends a hard reservation message requesting the same amount of bandwidth. This hard reservation will remove all the traffic occupying the corresponding soft reserved bandwidth. So after a hard reservation, the QoS traffic can immediately start running with its necessary QoS support. Introducing these two kinds of reservations is to achieve good performance in QoS monitoring.

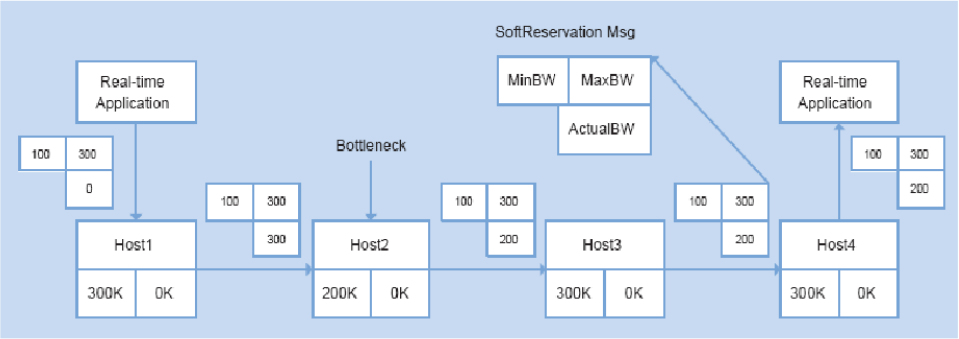


Figure 17 ASAP soft reservation

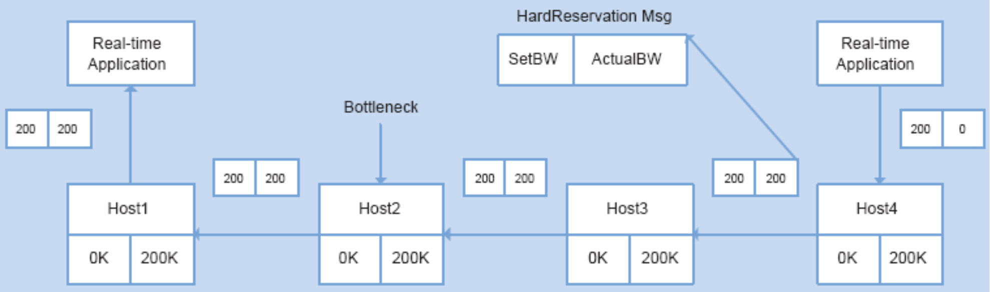


Figure 18: ASAP hard reservation (Stuedi et al., 2004).

Every node within the network stores information for each real-time flow having a reservation on that specific node. The per-flow information stored comprises a flowID uniquely identifying the flow and the actual soft and hard reservation for the flow. The set of all tuples stored within a node is called QoS table. Table updates are triggered upon receiving signalling messages (Stuedi et al., 2004) as illustrated in Table 7. SoftResv is the bandwidth soft reserved and HardResv is the hard reserved bandwidth.

Upon receiving this SR, the host considers it as a flow *setup request ()*, and switches the state to ESTABLISHING, indicating it is ready for flow setup. The host creates a flow entry in its flow table as shown in Table 7, makes a soft reservation within the range MinBW - MaxBW and marks the SoftBW field with the reserved amount. When the receiver gets the SR message, it knows that the available bandwidth equals the value of SoftBW. It then replies with an HR message with SetBW equal to the SoftBW in the SR message. This HR message is used as a method - flow *setup confirm ()*, and it travels back to the sender along the incoming path. Upon receiving the HR message, each intermediate host switches its soft reservation to hard reservation. It also marks the SoftBW/HardBW fields in the message, releases the extra reservation if it exists, and updates it flow table. Now the state of the node is switched from ESTABLISHING to ESTABLISHED, which means the node is ready to provide QoS support for that flow. Finally, when the HR message arrives at the sender, the sender can start the flow with a speed matching the reserved bandwidth. A big advantage of this approach is that the combination of soft and hard reservation helps to avoid wasting resources in hosts other than the bottleneck, a problem which other QoS protocols fail to deal with.

Table 7: QoS Table for ASAP

Flow Label	SrcAddress	SoftResv	HardResv
0	Host1	100	100
1	Host1	100	50
0	Host2	0	150

QoS monitoring packets periodically investigates the QoS situation on every node within a certain path. Hard reservation messages are sent whenever the end-to-end QoS changes. Monitoring interval can be changed dynamically. For example more frequent monitoring is needed, if the network is unstable, in order to adapt to bandwidth fluctuations. If the network is stable, processing overhead can be saved by keeping the monitoring rate low. ASAP also provides efficient in-band signalling for resource reservation, management, adaptation and releasing. The signalling is designed to produce minimum possible overhead and to provide maximum flexibility.

Problems of ASAP in MANETs

In his technical report (Stuedi et al., 2004) Patrick Stuedi explained the problems associated with his ASAP quality of service framework. The problems are listed and explained below.

a) Flow Restoration Problem

If a QoS path has been established from source to destination node and let’s assume that maximum quality of service is provided along this path. If at a certain time one node moves out of the others’ transmission range breaking the path, routing then finds a new path for the flow. Because no reservation is established in the new path the flow is transmitted using best effort. This state is kept until the next SR message detects the missing reservation and triggers a source node to send a hard reservation message, which will finally repair the reservation on the new path.

b) Reverse Path Problem

In ASAP, a hard reservation message is supposed to follow the reverse path that is previously established during soft reservation. This could be hard to achieve for several reasons. First, routes may change quickly in MANETs. A path established during soft reservation may be outdated while hard reservation is going on. Second, routes do not have to be symmetric. Although physically two nodes can reach each other in one hop distance that does not mean routing also behaves like this. This could result in a big latency for hard reservation messages. The other problem related to reverse paths occurs when wireless links are not symmetric. Even if a node A can reach B in one hop distance it is not given that node B is able to reach A as well. As a consequence there may be no way for a hard reservation to pass through.

c) Lost Hard-Reservation Messages

If a hard reservation message during adaptation gets lost after sending, no subsequent soft reservation message will trigger any hard reservation if the path condition (bandwidth allocations on the nodes) stays the same because the adaptation process already did update its bandwidth allocation value. This state is kept until the end-to-end bandwidth for the flow changes somehow, that means until a soft reservation message arrives at destination having an ActualBW value that is unequal to the one stored by the adaptation process. If no node is moving and bandwidth isn’t fluctuating either this may take a while. So a concept is needed to overcome this shortcoming. Hard reservation messages must be triggered until the reservation is actually done.

d) Incomplete differentiation

The differentiation problem of INSIGNIA and SWAN still appear in ASAP. ASAP differentiates traffic into Quality of Service and Best-effort traffic only. There is need for a better and more elaborate way of differentiating multimedia traffic according to some priorities depending on the bandwidth needs and maximum allowable delay for each kind of traffic.

2.7.8 CEQMM

Complete and efficient quality of service model for MANET (CEQMM) (Ibrahim, Mehmood, & Ullah, 2011) is a hybrid scheme, combining IntServ (per-flow) for traffic with highest priority and DiffServ (per-class QoS provisioning) for traffic with other priorities. Model consist of priority classifier, active queue management, packet scheduler and congestion avoidance mechanisms. In such a scheme, QoS traffic of highest priority is given per-flow provisioning while other priority QoS classes are given per-class provisioning. To offer this scheme and to ensure that certain packets receive higher priority transmission than other packets, priority classifier, active queue management and packet scheduler are integrated. CEQMM applies the QOLSR protocol to support multiple-metric routing criteria and to respond quickly when changes in topology and/or QoS conditions are detected. Once a path is chosen for one QoS flow, CEQMM performs call admission control (CAC) at each intermediate node. For only QoS flows of highest priority, a node can proceed to soft and later hard bandwidth reservation on links during the CAC process. CEQMM implements congestion avoidance mechanisms to prevent a network from entering the congested state. However, in MANETs, network congestion can still occur frequently under mobility. In order to prevent performance degradation due to mobility-triggered congestion, CEQMM uses a new congestion control scheme.

One limitation of implementing CEQMM for mobile Ad-hoc networks is that in case of continues movement of nodes, the average delay is around 400ms, which is the maximum delay allowed for most of the real-time applications that leads to more packets being dropped. This shows that real-time applications suffer significant transmission delays under the intense movement situations.(Ibrahim et al., 2011)

2.7.9 QoSMMANET

QoSMMANET (QoS Management in Mobile Ad hoc Networks) framework (Duran-Limon, Siller, Hernandez-Ochoa, Quevedo, & Robles, 2014), which offers QoS support for real-time event systems in highly mobile ad hoc environments. They expressed node mobility in terms of node velocity. The QoSMMANET framework uses the Probabilistic Flooding Protocol which is based on a flooding mechanism which limits packet redundancy. A QoSMMANET framework consists of the following three building blocks or modules as shown Figure 19:

i. Routing Protocol Block. This module is in charge of enabling end-to-end connectivity. This protocol is based on a probabilistic flooding mechanism. It is intended to cope with the network dynamics derived from node mobility whilst limiting network congestion.

ii. Traffic Differentiation: Queuing Discipline. This module is a mechanism oriented to provide packet differentiation and prioritisation. It supports two queuing methods: FIFO and WFQ (Weighted Fair Queuing).

iii. Bandwidth Allocation Protocol: QoS Management Protocol. The main goal of this module is to balance network load based on end-to end connectivity. Network traffic bottle necks are identified and traffic flows are regulated accordingly.

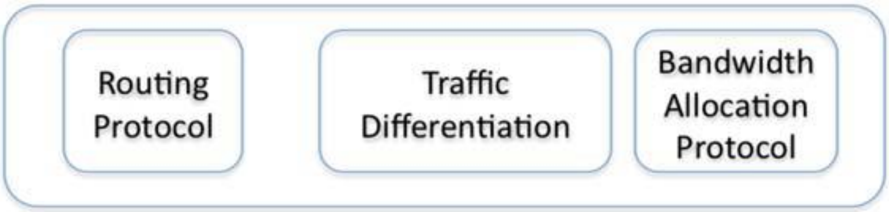


Figure 19: The QoSMMANET Framework.

QoSMMANET framework offers soft real-time QoS support. There is need for the framework to offer better than soft real-time QoS guarantees to hard real-time mobile ad hoc systems. Other issues can be addressed such as security, battery (energy) consumption, jitter, Normalized Routing Load (NRL), load balancing, and scalability. There is also a need to provide both a high probability of meeting deadlines and an adaptable and flexible infrastructure

2.7.10 iMAQ

The integrated Mobile Ad-hoc QoS framework (iMAQ) (Kandari & Pandey, 2014) is a cross-layer architecture to support the transmission of multimedia data over a MANET. The main idea of this model is based on a cross-layer communication approach involving network and application layer by means of so called middleware service layer. As nodes are mobile, the network can become partitioned which leads to missing data. A predictive location-based QoS routing protocol with middleware layer cooperation can predict network partitioning and provide necessary information to the application layer. Thus the main role of the middleware layer is to replicate data among different network groups in order to provide better data accessibility before network partitioning occurs. The disadvantage of this QoS model is its high overhead and lack of resource reservation.

2.7.11 DEQA

(Sulthani & Rao, 2009) discuss the Design of an Efficient QoS Architecture (DEQA) model. It consists of three parts, the routing protocol, admission control and congestion control. The routing protocol searches for several parallel communication paths. Data packet are fragmented in source node and traverse different paths to the destination where they are reassembled. The admission control define minimum and maximum thresholds. If the incoming flow’s QoS requirement is above the maximum limit, it is denied. On the other hand, if the requirement is below the minimum, it is allowed. If the QoS requirement is between minimum and maximum, a probe packet is sent along the communication path to the destination to gather information on available network resources upon which a decision of whether to allow or deny can be made. The congestion control periodically monitors the path for congestion. If congested, Explicit Congestion Notification (ECN) technique is used to decrease the transmission rate of the network traffic that does not require QoS provisioning.

Table 8: Comparative analysis of QoS models in litererature

Framework	**Techniques used**	**Disadvantages**
IntServ	• Provides per-flow guarantees end-to-end by reserving resources along the path. • Uses the Resource Reservation Protocol (RSVP) to reserve resources in each intermediate node, which requires an admission control for each node	• Scalability problem • Signalling problem
DiffServ	• QoS provisioning per aggregate class – DSCP (TOS IP header field) used to indicate traffic class – No e2e signalling – No per-flow state information maintained by interior routers ==> better scalability properties	• DiffServ is • based on the concept of SLA • Ambiguous core network
FQMM	• Hybrid per-flow (IntServ) and per-aggregate (DiffServ) QoS provisioning – Traffic divided into classes – Highest priority class given per-flow provisioning – Rest given per-class provisioning • Argument: per-flow states only needed for a subset of the flows, thus improving IntServ scalability	• scalability problem still exists • DiffServ Behaviour in ingress nodes.
INSIGNIA	• Uses piggybacked signalling packets • Frameworks allows also	• Scalability problem due to the flow state information which is kept within the

	realtime applications to specify minimum and maximum bandwidth needs	nodes of a certain path. • Bandwidth usage in INSIGNIA is not efficient • No mechanism to dynamically change the frequency by which control signals are inserted into the data packets. • Only two bandwidth levels to be used, MINIMUM and MAXIMUM.
SWAN	• A stateless network QoS model which uses distributed control algorithms with additive increase multiplicative decrease (AIMD) rate control mechanism to deliver service differentiation in mobile wireless ad-hoc networks. • The SWAN model includes a number of mechanisms used to support rate regulation of BE traffic and admission control regulation of RT traffic	• It can only provide weak service guarantees • source-based admission control using probing packets is again unrealistic and ineffective in a MANETs • bandwidth calculations in SWAN do not take best effort traffic into consideration
ASAP	• An adaptive reservation QoS protocol. • By adopting a simple signalling system and a two-phase reservation mechanism, ASAP provides adaptive QoS support, fast flow path setup and local repairing, as well as processing optimization.	• Flow Restoration Problem • Reverse Path Problem • Lost Hard-Reservation Messages
IMAQ	• A cross-layer architecture to support the transmission of multimedia data over a MANET.	• This QoS model has high overhead and lacks resource reservation

	• They use a location-based pro-active QoS-Routing. Neither hard QoS guarantees can be provided nor are any resources reserved.	
INORA	• A QoS support mechanism that makes use of the INSIGNIA in-band signalling and TORA routing protocol for MANETs. • INORA represents a QoS-signalling approach in a loosely coupled kind of manner. • The idea is based upon the property of TORA to provide multiple routes between a given source and destination. • INORA gives feedback to the routing protocol on a per-hop basis to direct the flow along the route that is able to satisfy the QoS requirements of the flow.	• The shortcomings of INORA mostly are the shortcomings of INSIGNIA. • However, the interface for signalling to access routing should be as generic as possible in order to guarantee portability.

CEQMM	• Uses a hybrid scheme by allowing both per-flow and per-class provisioning of services to the mobile nodes in the mobile Ad-hoc network. • Uses the QOLSR protocol for supporting multiple-metric routing criteria and at to respond quickly in case of topological change. • The CEQMM also provides a mechanism for congestion avoidance to avert the network from entering into a congested state.	• In case of continues movement of nodes, the average delay is around 400ms, which is the maximum delay allowed for most of the real-time applications that leads to more packets being dropped
DEQA	• It consists of three parts, the routing protocol, admission control and congestion control. • The admission control define minimum and maximum thresholds. • A probe packet is sent to the destination to gather information on available network resources for admission control	• Scalability problem due to the flow state information which is kept within the nodes of a certain path • Lost QoS probe packets

QoSMMANET	• Node mobility is expressed in terms of node velocity. • It uses the Probabilistic Flooding Protocol which is based on a flooding mechanism which limits packet redundancy.	• It offers only soft real-time QoS support. There is need for the framework to offer better than soft real-time QoS guarantees to hard real-time mobile ad hoc systems. • Other issues include security, battery, consumption, jitter, Normalized Routing Load (NRL), load balancing, and scalability. • There is also a need to provide both a high probability of meeting deadlines and an adaptable and flexible infrastructure

2.8 Conclusion

In this chapter we defined quality of service, and identified the factors affecting quality of service in mobile ad hoc networks. We went ahead and described the quality of service metrics and factors that affect them in mobile ad hoc networks. We looked at quality of service models already in existence for mobile ad hoc networks. These include IntServ, DiffServ, FQMM, INSIGNIA SWAN and ASAP. We identified that the IntServ and DiffServ were designed for the wired network so they do not quite fit for mobile ad hoc networks. Whilst FQMM is a good model which combines the strengths of IntServ and DiffServ, it also carries most of their disadvantages with it which makes it not quite suitable a model for mobile ad hoc networks. INSIGNIA is a well-designed signalling approach for MANETs it exhibit some inherent problems. These drawbacks of INSIGNIA are its scalability problem due to the flow state information, which is kept within the nodes of a certain path and inefficient bandwidth usage. The bandwidth management of SWAN, though very good it is not very good for MANETs since it is not a complete QoS solution for a highly dynamic network like a MANET. Although ASAP makes use of in-band signalling and fast adaptation but the protocol still fails to meet some MANET specific demands. Few problems of ASAP in a mobile ad hoc environment include flow restoration problem, reverse path problem and lost Hard-Reservation messages. All and above all the problems associated with all these quality of service models described in this chapter, there is need for a model that can do traffic classification that takes into account different types of traffic that make up multimedia traffic and their varying bandwidth requirements. It does not make sense to give all real-time traffic the same priority since they come with different bandwidth, throughput and delay needs and therefore they must be treated differently according to their needs.   The model should also be able to do good bandwidth management based on an intelligent adaptation method that recognizes the priorities of the traffic. For this to be possible, the first thing should be an excellent bandwidth estimation method that makes the base of the management system. The next chapter will discuss a bandwidth estimation method for MANETs.

3. Bandwidth Estimation[2]

3.1. Problem Definition

Bandwidth is a scarce resource in most wireless networks and more so for mobile ad hoc networks. For effective transfer of packets from the source to destination, there should be enough bandwidth in the path. With the current increase in the interest of the use of multimedia traffic in wireless mobile ad hoc networks, proper and accurate bandwidth management is of paramount importance. Measuring the link capacity and the available bandwidth on the network is important in the transmission of multimedia applications such as videoconferencing and voice over IP. Multimedia traffic normally requires knowledge of both the link capacity and the available bandwidth, in order to make decisions, such as admission control and bandwidth management.

Available bandwidth estimation is important for successful admission control and bandwidth management in computer networks. Knowing the available bandwidth can help in developing protocols to create and maintain quality of service through traffic engineering, channel selection, admission control and bandwidth adaption and routing. Quality of Service aware routing depends very much on the accuracy of the bandwidth availability at network nodes and the path between the sending and receiving pair. QoS aware routing uses this information to find paths that can satisfy certain QoS requirements. The Admission Control scheme is an important component of a network for providing QoS assurances. For bandwidth management to be successful, then bandwidth estimation should be accurate also. Knowledge of the amount of bandwidth is usually obtained by estimation rather than by measurement.

Bandwidth estimation in wired networks is usually easy but in wireless networks it is difficult as a result of various problems associated with the character of wireless networks. The first problem is due to interference between nodes.

[2] This chapter is an extension of work published in:
B. M. Nyambo, G. K. Janssens, W. Lamotte , Bandwidth Estimation in Wireless Mobile Ad Hoc Networks, *Journal of Ubiquitous Systems & Pervasive Networks, Volume 6, No. 2 (2015) pp. 19-26*

Because nodes share the wireless channel, the available bandwidth varies with the number of nodes in the network. Nodes within transmission range of each other will always share the same channel and compete for channel usage. The second challenge is due to channel fading. Channel fading and error from physical obstacles cause the available bandwidth to experience fast time-scale variations. The other challenge comes because of the constant mobility of nodes which always makes bandwidth estimation a problem since it can change once a node moves out of range breaking transmission paths or a new node comes into interference of the other causing interference and possible congestion. Several approaches have been proposed in literature but there is little or no consensus on the best or precise method of measuring the available bandwidth. It is thus so important to rethink the available bandwidth estimation in mobile ad hoc networks and articulate the challenges associated therewith.

In this chapter we have various objectives with the endeavour of designing methods of estimating available bandwidth in mobile ad hoc networks. The first objective is to study various bandwidth estimation methods available in literature. The second objective is to propose a new bandwidth estimation method building on the knowledge obtained from strengths and weaknesses of methods in literature. A new algorithms will be developed to solve the problem of estimating bandwidth in MANETs. The third objective is to do a performance modelling of this algorithm and evaluate is effectiveness in estimating bandwidth in MANETs.

3.2. System definition

Consider a network path shown in Figure 20. The data packets will have to traverse the path from node *N1* up to node N_H by making sequential hops through links starting with *Link*$_1$ up to *Link*$_{H-1.}$

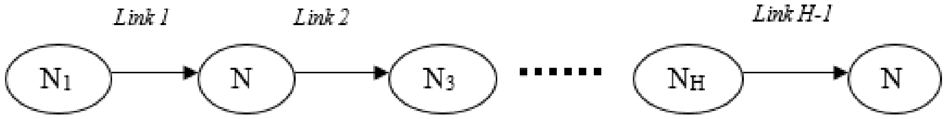


Figure 20: An H-1 hop path to calculate available bandwidth

In such a path the channel capacity C was defined in (Prasad & Murray, 2003) as:

$$C = \min_{i\,=\,1..H} C_i \qquad 3.1$$

In this case, C_i is the link capacity of the i^{th} hop in an *H-1* hop path. This means that the capacity is determined by the bottleneck link capacity, which is the link with the small value in magnitude.

The state of a *Link i* at time *t* can be defined to be 0 when the link is idle and 1 when the link is busy. A link is said to be busy when it is transmitting, receiving or sensing some interference from neighbouring nodes. Therefore, the average utilization of a *Link i* can be expressed as:

$$u^{\tau}(t) = \frac{1}{\tau}\int_{t-\tau}^{t} u(x)\,dx \qquad 3.2$$

where $u(x)$ is the instantaneous utilization of the link at time x and τ is the averaging time-scale. If we define the available bandwidth $AB_i\,(t)$ of a Link i as the unused bandwidth on that link over a period$[\,t-\tau, t]$, then we can express AB_i as:

$$AB_i(t) = C_i\big(1 - \, u^{\tau}(t)\big) \qquad 3.3$$

Thus the available bandwidth AB for the path $1 \cdot\cdot H$ over the same period can be expressed as:

$$AB = \min_{i=1..H} A_i \qquad 3.4$$

Therefore, the available bandwidth is determined by the hop with the minimum amount of available bandwidth. This is sometimes called (Prasad, 2003) the bottleneck link or the tight link of the end-to-end path. For a new traffic flow to be admitted in a network it will depend on the available bandwidth in a path and is not dependent on the link capacity. Link capacity may only affect the number of flows that a link can sustain.

In a MANET, a host's available bandwidth is decided by the raw channel bandwidth, its neighbour's bandwidth usage and interference caused by other sources. Each of these elements reduces a host's available bandwidth. Therefore, applications must have knowledge of the entire network for them to properly optimize their coding rate. This implies that computation of the available bandwidth between two neighbour nodes requires identification of all the emitter's potential contenders and of all the receiver's potential jammers. These nodes' utilization of the shared resource should then be gathered and should be composed to derive the amount of free resources. This first means that a precise identification of all interfering nodes is required. Secondly, information on their shared bandwidth usage has to be gathered. Finally, the joint impact of the aggregated traffic should be evaluated. These tasks are usually hard to realize and they get even harder in sparse networks, as two nodes may interact without being able to exchange information.

3.3. Bandwidth Estimation Techniques

Several approaches can be distinguished that allow determining the available capacity. These can be classified into model based, measurement based and calculation based approaches. In model based bandwidth estimation approaches, researchers try to model the network using mathematical models which they use to predict network metrics like throughput, delay and packet drop ping probability. A good example is the Bianchi model (Bianchi, 2000) which was used to predict saturation throughput of IEEE 802.11 based WLAN networks. The model also calculates the probability of a packet transmission failure due to collision.

3.3.1 Model based techniques

In (Manshaei & Hubaux, 2007a) the model is explained very well in a summary. The model is based on the carrier sense multiple access with collision avoidance (CSMA/CA) protocol of the IEEE 802.11 MAC layer. The basic access method for the IEEE802.11 is called the distributed coordination function (DCF) and it uses the CSMA/CA algorithm to regulate the access to the shared medium. DCF includes the ready-to-send (RTS) and clear-to-send (CTS) access techniques as shown in Figure 21.

If a source node needs to transmit data, it checks the channel if any other node in its interference range is transmitting. If another node is using the channel then the source node will back-off for a random time chosen randomly in the interval [0, CW] where CW stands for contention window. The backoff time is measured in slot times. A slot time is defined as the sum of the Receiver-to-Transmitter turnaround time, MAC processing delay, and clear channel assessment (CCA) detect time.

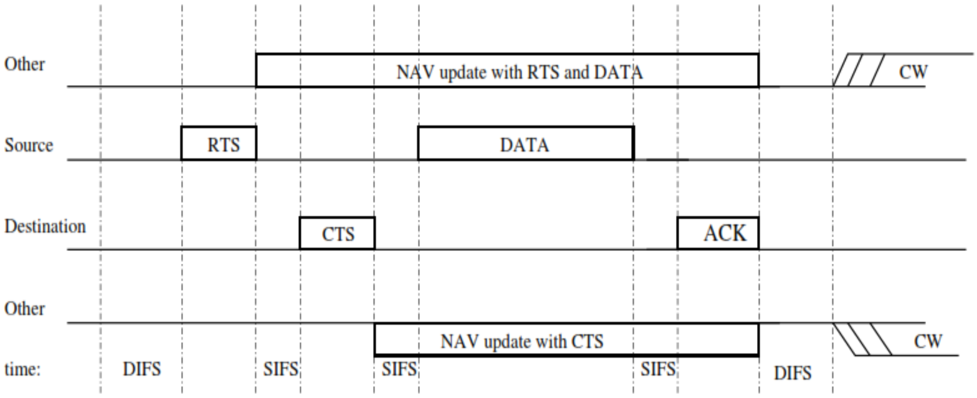


Figure 21: CSMA/CA protocol with RTS/CTS exchange mechanism

In short a slot time is the time required by a node to sense an end of frame, start transmitting and begin the frame to propagate to other nodes.

The DCF IEEE 802.11 describes two types of times, the Short Inter-Frame Space (SIFS) and the DCF Inter-Frame Space (DIFS).

The SIFS is the time required for a node to sense end of frame and start transmitting. DIFS is the time required for a node to wait before starting a back-off interval and can be expressed as,

$$DIFS = SIFS + 2 * \; SlotTime.$$

The back-off timer counts slots until the node’s turn to transmit. It is decreased by one every time the channel is found to be idle for a time equal to a DIFS. If the channel is busy the timer is stopped until the channel is idle again for at least a DIFS period.

The contention window is an integer of which the value is determined by physical layer characteristics called CW_{min} and CW_{max}. CW is doubled after every unsuccessful transmission up to a maximum value equal to CW_{max} + *1*. (Bianchi, 2000) defined $W_i = 2^i W$, where $i \in (0, m)$ which is the backoff stage and W is the 802.11 parameter CW_{min}. They later on defined a Markovian state *B(t) ={s(t), b(t)}* such that *b(t)* is the backoff time counter at time t and s(t)is the backoff stage (0, … ,m) at time t.

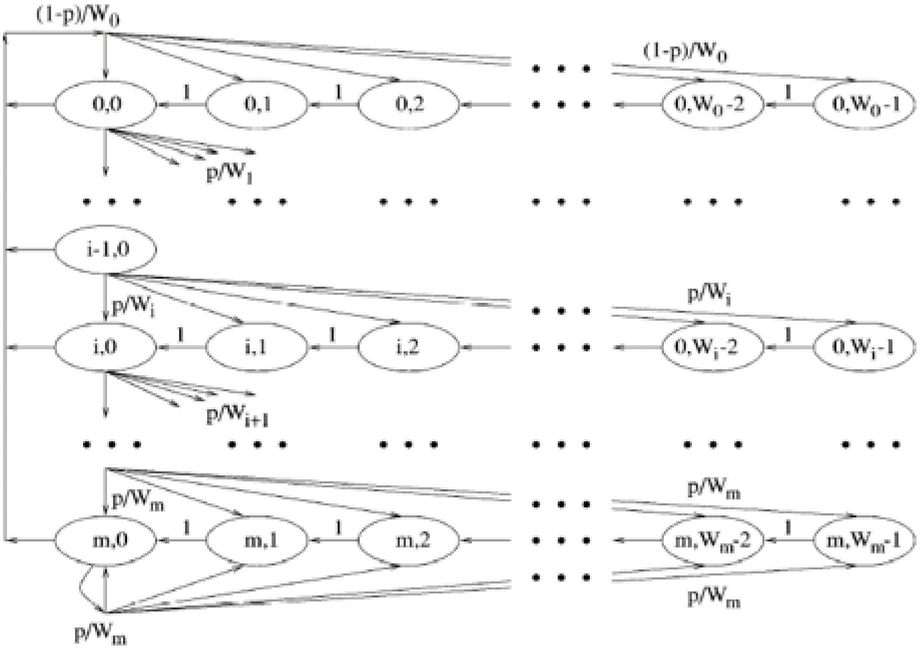


Figure 22: One step transition Markov Chain model (Bianchi, 2000)

The state *B(t)* is depicted in Figure 22 showing one-step transition probabilities of successful transmissions. The transition probabilities are expressed as:

.

$$\begin{cases} P\{i,k|i,k+1\} = 1 & k \in (0, W_i - 2); i \in (0, m); \\ P\{0,k|i,0\} = \dfrac{1-p}{W_0}, & k \in (0, W_0 - 1); i \in (0, m); \\ P\{i,k|i-1,0) = \dfrac{p}{W_i} & k \in (0, W_i - 1); i \in (1, m); \\ P\{m,k|m,0\} = \dfrac{p}{W_m} & k \in (0, W_m - 1). \end{cases}$$

In the same research they came up with τ, the probability for a station to transmit in a generic slot time.

$$\tau = \frac{2(1-2p)}{(1-2p)(W+1)+pW(1-(2p)^m)} \qquad 3.5$$

From this they defined a probability p that in a time slot, at least one of the $n-1$ remaining stations transmits.

$$p = 1-(1-\tau)^{n-1} \qquad 3.6$$

From these two equations the values of τ and p can be found and the probability, P_{tr} and P_s can be found. P_{tr} is the probability that there is at least one transmission in a slot time and P_s is the probability that a transmission is successful.

$$P_{tr} = 1-(1-\tau)^n$$

$$P_s = \frac{n\tau(1-\tau)^{n-1}}{P_{tr}} = \frac{n\tau(1-\tau)^{n-1}}{1-(1-\tau)^n} \qquad 3.7$$

The saturation throughput, which is the average information payload transmitted in a slot time over the average duration of a slot time, can be computed as follows (Manshaei & Hubaux, 2007b):

$$S = \frac{E[Payload\ information\ transmitted\ in\ a\ slot\ time]}{E[Duration\ of\ slot\ time]}$$

$$= \frac{P_S P_{tr} L}{P_S P_{tr} T_S + P_{tr}(1-P_S)T_c + (1-P_{tr})T_{id}} \qquad 3.8$$

Variable T_s is the average time needed to transmit a packet of size L (including the inter-frame spacing periods; T_{id} is the duration of the idle period (a single slot time); and T_c is the average time spent in the collision.

In his work (Bianchi, 2000) made some assumptions, some of which include:

(1) Every node is saturated
(2) Transmission error is a result of packet collision and not caused by channel errors
(3) The network is homogeneous (that is each node acts in the same way)

These assumptions are bound to be invalid in a real mobile ad hoc network. This will definitely limit the accuracy of the model. To take care of these problems (Zhao, Wang, Wei, Song, & Li, 2011) came up with a new analytical model based upon the sender – receiver (S-R) pair model. This model gives the link state from the view of the S-R pair, takes network information, radio dependent parameters and incoming traffic throughput demands as inputs, and gives the predictive throughputs of both ongoing traffic and incoming traffic. For a link k, they described four time intervals when the link is in the idle state (σ) successful transmission (T_k), unsuccessful transmission/ collision (C_k) and channel busy (B_k) and they described a generic slot of length E_k , expressed as:

$$E_K = \tau_k \,.\, p_k . C_k + \tau_k . (1 - p_k). T_k + (1 - \tau_k). b_k . B_k + (1 - \tau_k). (1 - b_k). \sigma \qquad 3.9$$

The variables τ_k , p_k and b_k are the probabilities of transmission, unsuccessful transmission and channel sensed busy on one time slot. For a generic link k, (Zhao et al., 2011) came up with expressions for the channel utilization ratio(x_k), successful transmission ration (f_k) , and the throughput(S_k) given an effective load fraction.

The channel utilization ratio is the ratio of the time the channel is determined to be busy to the total time. It is given by the periods of successful transmissions as well as collisions.

$$x_k = \frac{\tau_k .p_k .C_k+(1-p_k).T_k}{E_k} \qquad 3.10$$

The successful transmission ratio is the ratio of successful transmission periods to the total time. It counts all the periods with successful transmission including the times for RTS, CTS, DATA, ACK and all the inter frame spaces SIFS and DIFS.

$$f_k = \frac{\tau_k .(1-p_k).T_k}{E_k} \qquad 3.11$$

Throughput can then be found from a relationship between the successful transmission ratio and the load in the network.

$$S_k = \frac{\tau_k .(1-p_k).\alpha}{E_k} \qquad 3.12$$

(Zhao et al., 2011) proposed an analytical model, Figure 23, which consists of three major components: The S-R (i.e., sender-receiver) pair model, the interference model and the bandwidth requirement mapping model. The S-R pair model gives the link state from the view of an S-R pair; the interference model constructs the contention graph of the network, in order to analyse the interference of contending links; the bandwidth requirement mapping model relates the network parameters in the S-R pair model and interference model to the bandwidth requirement of the incoming flow(s).

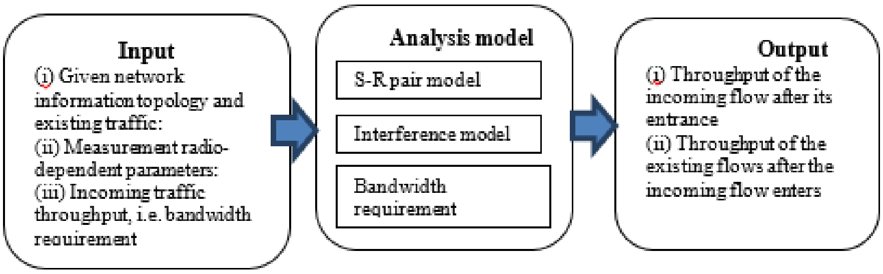

Input
(i) Given network information topology and existing traffic:
(ii) Measurement radio-dependent parameters:
(iii) Incoming traffic throughput, i.e. bandwidth reouirement
Analysis model
S-R pair model
Interference model
Bandwidth requirement
Output
(i) Throughput of the incoming flow after its entrance
(ii) Throughput of the existing flows after the incoming flow enters

Figure 23: Model based bandwidth estimation analytical model

For a given sender and receiver pair N_{k-1} and N_k, connected by Link *k,* four different states can be identified: idle (σ), successful transmission (T_k), unsuccessful transmission (C_k) and sense busy (B_k). Adopting the concept of a generic slot, the average length of the generic slot of link *k* can be expressed as

$E_k = \tau_k \cdot p_k \cdot C_k + \tau_k \cdot (1 - p_k) \cdot T_k + (1 - \tau_k) \cdot b_k \cdot B_k + (1 - \tau_k) \cdot (1 - b_k) \cdot \sigma$ 3.13

where τ_k, p_k and b_k respectively represent the probabilities of transmission, unsuccessful transmission and channel sensed busy in one time slot.

Then the normalized channel utilization ratio (x_k), can be expressed as

$$x_k = \frac{\tau_k \cdot p_k \cdot C_k + \tau_k \cdot (1 - p_k) \cdot T_k}{E_k} \qquad 3.14$$

The successful transmission time ratio (f_k) and the throughput (S_k) of Link *k* can also be expressed as:

$$f_k = \frac{\tau_k \cdot (1 - p_k) T_k}{E_k} \qquad 3.15$$

$$S_k = \frac{\tau_k \cdot (1 - p_k) \cdot \alpha}{E_k} \qquad 3.16$$

where α is the effective load fraction.

A node’s busy period is less than the sum of its contending nodes’ transmitting duration. It can be approximated as follows:

$$B_k = \left(\sum_{i \in v(k)} x_i - \sum_{\substack{i_1 \cdot i_2 \in v(k): \\ i_1 \notin v(i_2) \cup i_2}} \frac{x_{i_1} \cdot x_{i_2}}{1 - \sum_{c \in v(i_1, i_2)} x_c} \right) E_k \qquad 3.17$$

where *v*(*k*) represents the set of contending links of Link *k* and *v*(*i, j*) the set of common contending links of link *i* and link *j*.

They went on to define the transmission probability τ which is a function of unsuccessful transmission probability p, which is given by:

$$\tau = \eta \cdot \left(\frac{q^2 \cdot W_0}{(1-q)\cdot(1-p)\cdot(1-(1-q)^{W_0})} - \frac{q^2 \cdot (1-p)}{1-q} \right) \qquad 3.18$$

where η is the stationary probability of a node being in the state where the backoff process is complete, but the node's transmission queue is empty, q is the probability that there is at least one packet in the queue after a transmission and W_0 is the node's minimum contention window. In this research, they went on to define their Bandwidth requirement-mapping model. To satisfy the application bandwidth requirement (*Bw, bps*), expressed in bits per second (*bps*), given the traffic packet size (*PS, bits*), the packet arrival rate is

$$\lambda = \frac{B_w}{PS \cdot \alpha} \qquad 3.19$$

According to (3.12), they obtained that the transmission probability used for this application by a link (Link *k*) along its path is at least

$$\tau_k = \frac{Bw \cdot E_k}{PS \cdot (1-p_k) \cdot \alpha} = \frac{\lambda \cdot E_k}{(1-p_k)} \qquad 3.20$$

The coupling of the network parameters relates the bandwidth requirement of flows to all the network parameters.

3.3.2 Measurement based approaches

In measurement-based approaches, researchers apply active measurement techniques to estimate available bandwidth or capacity of a link or a path. Several measurement tools were developed, mostly based on sending probing packets, which can create significant overhead and severely interact with the real data currently transmitted in the network. In this section we will describe a few examples of measurement based approaches to bandwidth estimation.

3.3.2.1 Self-Loading Periodic Streams (SLoPS)

Self-Loading Periodic Streams (SLoPS) is described in (Jain & Dovrolis, 2002). In this method, a source node sends a periodic packet stream of approximately 100 packets of the same size and predetermined rate *R*. The one way delays of these probing packets are measured and if the rate R exceeds the available bandwidth, then the delay starts to increase. The receiving end sends to the sending node values of the one way delays of the packets. The sender sends theses streams of packets separated by silent periods such that the traffic from probing does remain below 10% of the total traffic. It also makes sure that there is no more than one packet stream in the network at each instance.

The sending node will try to bring the sending rate R close to the available bandwidth using an iterative algorithm. If the sending rate is lower than A, then packets will go through the network without causing an increase in delay. However, if the rate R is above A, then the one way delay will progressively increase.

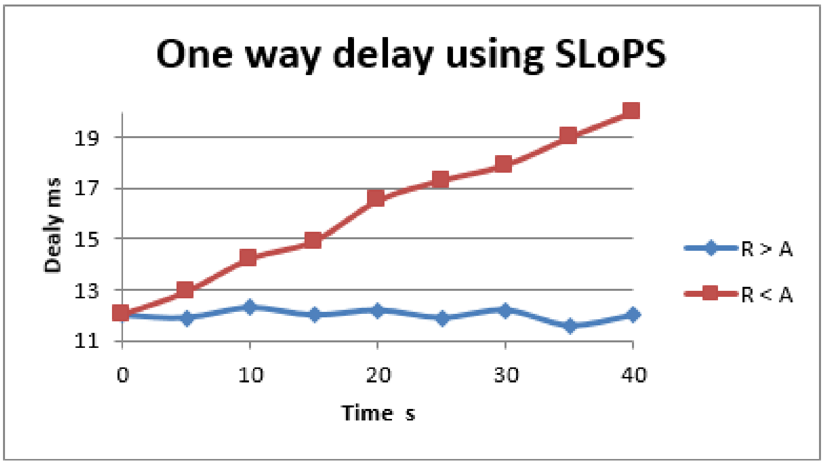


Figure 24: One-way delay for two streams using SLoPs

Figure 24 shows the behaviour for two streams of packets under two conditions (R > A) and (R < A) and the response of the one way delay. If the available bandwidth estimate A varies during measurement, SLoPS detect this by noticing that the one way delays do not show a clear increasing or non-increasing trend.

In such a case a grey region is reported which is related to the variation range of A during the measurement.

Trains of Packet Pairs (TOPP), another method similar to the SLoPS, was proposed by (Melander, Francisco, & Engineers, 2000) but it uses a pair of packets to probe the network. Many packet pairs are sent from source node to destination node gradually increasing the sending rate each time.

3.3.2.2 The probe gap model (PGM)

The PGM (Strauss et al., 2003) uses the time gap between the arrival times of two successive probe packets at the receiving end. A pair of probing packets is sent with a time gap Δ_{in}, and reaches the receiver with a time gap Δ_{out}. If we assume that there is one bottleneck, then Δ_{out} is the time taken by the bottleneck to transmit the second probe in the pair and the cross traffic that arrived during Δ_{in} as shown in Figure 25. Thus, the time to transmit the cross traffic is Δ_{out} - Δ_{in}, and the rate of the cross-traffic is $\frac{\Delta_{out}-\Delta_{in}}{\Delta_{in}} \times C$, where C is the capacity of the bottleneck. The available bandwidth is:

$$A = C \times \left(1 - \frac{\Delta_{out}-\Delta_{in}}{\Delta_{in}}\right) \qquad 3.21$$

Spruce, IGI, and Delphi are example tools that use the gap model.

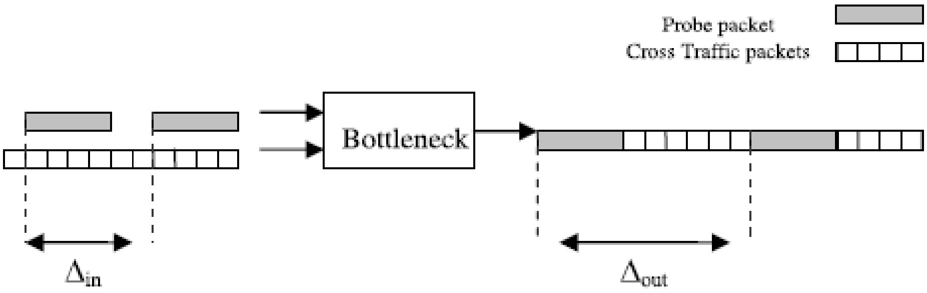


Figure 25: The Probe Gap Model for estimating bandwidth.

3.3.2.3 The Probe Rate Model (PRM)

The probe rate model (PRM) is based on the concept of self-induced congestion; if a source node sends probe packets at a rate lower than the available bandwidth along the path, then their arrival at the destination will match their rate at the destination. However, if the rate of probe packets traffic is higher than the available bandwidth, then the packets will queue at destination. Thus, the available bandwidth is measured by searching for the turning point at which the probe sending and receiving rates start matching. The PRM model has been used in many available bandwidth estimation tools and it has been shown as very accurate. However it suffers from one basic problem: PRM based tools must send probe traffic at a rate equal or greater than the AB. This will fill the queues along the path congesting it. This means that, for each estimation, a PRM-based tool congests the measured path during a certain period of time. In fact tools such as Pathload, Pathchirp, and TOPP which use the probe rate model can significantly impact the response time of TCP connections.

Jacobson of the Network Research Group developed a tool called Pathchar (Jacobson, 1997). This is a tool to characterize the bandwidth, latency, and loss of links along an end-to-end path through the Internet. Although developed back in 1997, Pathchar remains relevant for modern use due to its underlying principles. Its main goal is to enhance traceroute such that more information is obtained and returned to the user. Traceroute is a network diagnostic tool for displaying the path and measuring transit delays of packets across the network. During execution, pathchar sends and tracks a specified number of packets with a variety of sizes to a destination, returning the bandwidth, propagation delay, round-trip time, and queuing delay for each hop based upon the average values obtained by the packets. While this program provides a significant amount of path and bandwidth information to the user, it does have its drawbacks. One issue is that it does not correctly analyse paths that respond significantly differently to packets of different sizes. Additionally, the utility requires customization in almost a trial-and-error method to determine appropriate sample sizes for each network.

3.3.2.4 Pathload

Pathload tool calculates the available bandwidth on a path, as defined by the amount of traffic that a source can generate without disrupting the other data transfers on the links. This tool is based upon one central principle: if a stream of User Datagram Protocol (UDP) packets is sent at a rate that exceeds the available bandwidth, packet delays will increase along that transmission path. However, if the available bandwidth is not met, then these delays will have no discernible trend. Pathload uses this knowledge by sending differing sizes of test streams to approximate the available bandwidth. This can be done successively to upper-bound the range of bandwidth values with a very high degree of precision. However, one major issue exists: if the normal traffic has non-trivial changes in intensity over time, Pathload's bounding may be inaccurate or may not converge. Although the results obtained from Pathload may be interesting and useful for stable networks, unless bandwidth variation is consistently quite low, they cannot be reliably used for predictive analysis.

3.3.2.5 Pathrate

Pathrate is an end-to-end capacity estimation tool, created by (Dovrolis, Ramanathan, & Moore, 2004). This method tries to find the bottleneck and maximum possible bandwidth of a path. Restarting this, calculations are made to determine the highest bandwidth possible, rather than the current bandwidth available. Pathrate sends a long packet train consisting of dozens of groups of packets of different sizes, in order to obtain a large sampling of data concerning operational bandwidth. This data is taken and averaged to give a relatively narrow bounding of path capacity, quantified by a returned coefficient of variation. Although the statistical techniques used within this paper are valid, the range of path bandwidths can only be stated with a high degree of reliability, not absolute certainty. Therefore, it would be prudent to run Pathrate multiple times before attempting to send at a rate near the specified maximum.

It uses packet-pairs and packet-trains (`packet dispersion' methods), in conjunction with statistical techniques, to estimate the capacity of the narrow link in the path. Pathrate requires the user to have access at both ends of the path. It uses UDP packets for the packet-pairs and packet-trains, as well as a TCP connection for exchange of control information. It can be run from user-space, and it does not require superuser privileges. Pathrate operates in three phases. In the Initial Phase, it sends a few packet trains of increasing length to detect if the narrow link has parallel sub-channels, or if it performs traffic shaping. In the next phase, it generates thousands of packet-pairs of variable size, in order to make the non-capacity local modes weaker and wider. In the final phase, Pathrate estimates the Asymptotic Dispersion Rate (ADR) and the capacity estimate for the path.

3.3.2.6 PathChirp

PathChirp (Ribeiro, Riedi, Baraniuk, Navratil, & Cottrell, 2003) is an active probing tool for estimating the available bandwidth on a communication network path. It is based on the concept of "self-induced congestion." PathChirp sends successive groups of packets, called chirps, each of which increases in the number of probes per group and then conducts a statistical analysis at the receiver. PathChirp is able to observe packet interarrival times and then averages results in order to find the minimum and maximum possible bandwidth. By rapidly increasing the probing rate within each chirp, PathChirp obtains a rich set of information from which to estimate the available bandwidth, dynamically.

3.3.2.7 Trains of Packet Pairs (TOPP)

(Melander et al., 2000) proposed a measurement methodology to estimate the available bandwidth of a network path, called TOPP. TOPP sends many packet pairs at gradually increasing rates from the source to the sink. Suppose a packet pair is sent from the source with initial dispersion ΔS. The probing packets have a size of L bytes; thus, the offered rate of the packet pair is $Ro = L/\Delta S$. If Ro is more than the end-to-end available bandwidth A, the second probing packet will be queued behind the first probing packet, and the *measured rate* at the receiver will be $R_m < R_o$. On the other hand, if $R_o < A$, TOPP assumes that the packet pair will arrive at the receiver with the same rate it had at the sender (i.e., $R_m = R_{o.}$ To illustrate TOPP Figure 26, (Prasad & Murray, 2003), consider a single-link path with capacity C, available bandwidth A, and average cross traffic rate $R_c = C – A$.

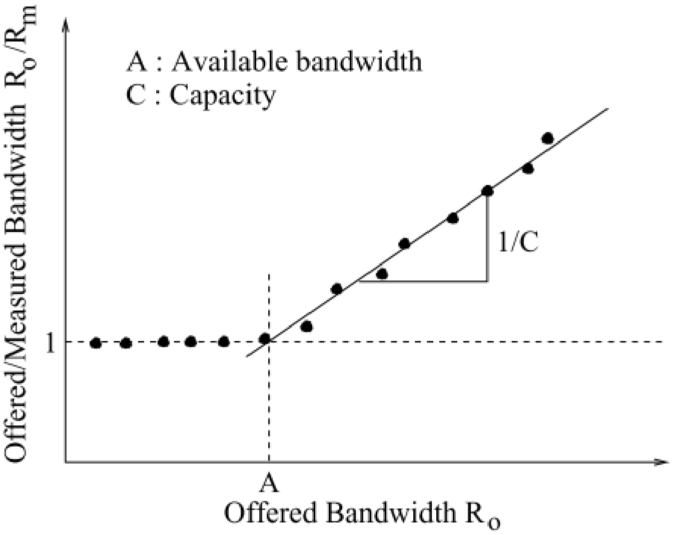


Figure 26: Offered bandwidth over measured bandwidth in TOPP

TOPP sends packet pairs with an increasing offered rate *Ro*. When *Ro >A*, the measured rate of the packet pair at the receiver will be

$$R_m = \frac{R_0}{R_0+R_C}C \qquad 3.22$$

TOPP estimates the available bandwidth *A* to be the maximum offered rate such that $R_o \approx R_m$. Equation 3.19 estimates the capacity C from the slope of R_o/R_m vs. R_o. In paths with multiple links, the R_o/R_m curve may show multiple slope changes due to queuing at links having higher available bandwidth than *A*. Unfortunately, the estimation of bandwidth characteristics at those links depends on their sequencing in the path.

3.3.2.8 DietTopp

(Johnsson, Melander, Björkman, & Bjorkman, 2004) designed another tool based on TOPP, called DietTopp. DietTopp injects a set of *m* probe-packet trains at an increasing rate in the interval $[o_{min}, o_{max}]$. On the receiver side each probe packet is time stamped in order to calculate m_i for each incoming probe-packet train *i*. The probe-packet train rate increases for each successive train, hence the bottleneck link will be congested at some point (corresponding to 0_i = *a*, in Figure 27). When all probe-packet trains have traversed the network path the quotient o_i/m_i can be plotted in the *y axis*. The rate response curve in Figure 27 is used as an example. DietTopp uses linear regression to estimate the linear segment *b*. The end-to-end available bandwidth is defined as the offered rate corresponding to the intersection of *b* and *y = 1*. Further, the slope of b corresponds to the bottleneck link capacity according to.

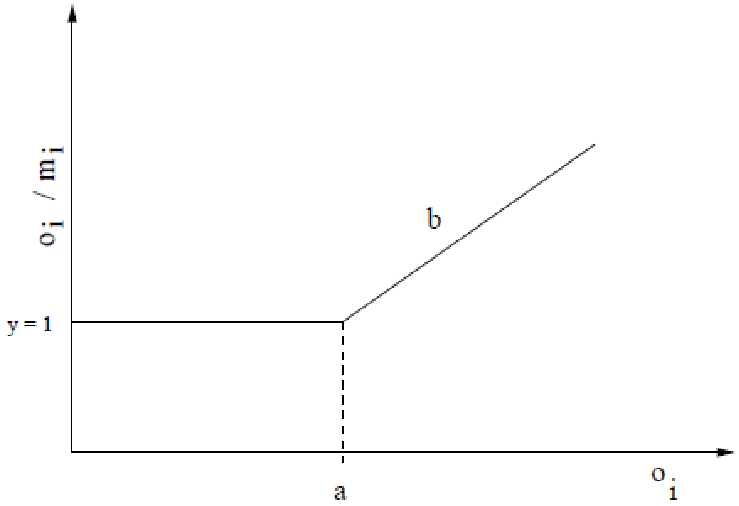


Figure 27: Plot of the ratio o_i/m_i as a function of o_i.

To speed up the probing phase of DietTopp it is desired to avoid measurements with an offered rate O below a. That is, DietTopp wants to ensure that $O_{min} > a$. This is done by estimating m_{max} which is done by injecting a set of probe packets at rate o_{max} (could be the link capacity of the access link for example) and then measure their separation at the receiver. The value m_{max} is greater than the available bandwidth and is referred to as the asymptotic dispersion rate. DietTopp also assumes only one bottleneck link between the end nodes contrary to the more computational expensive TOPP model.

The authors showed that the probe packets size as well as the volume of cross-traffic have a stronger impact on the measured bandwidth than in wired networks. Aside from the fact that this only measures the path capacity rather than the available bandwidth, this indicates that these techniques lead to inaccurate results in wireless networks.

3.3.2.9 WBest

(Li, Claypool, & Kinicki, 2008) came up with WBest, a bandwidth estimation tool for multimedia streaming over IEEE 802.11 Wireless Networks. It calculates the effective capacity, the achievable throughput, and the available bandwidth, on a network path over the network. WBest has been shown to offer much more accurate bandwidth estimations compared to three popular tools proposed for the Internet. Compared to other tools for wireless networks, it is very fast, using only one train of 30 packets sent at a rate equal to the effective capacity to estimate the achievable throughput and the available bandwidth (in contrast, e.g., DietTopp sends a series of packet trains, each one at a different rate), and it calculates all three bandwidth-related metrics (in contrast, e.g., ProbeGap only estimates available bandwidth but it requires a priori knowledge of the capacity).

WBest uses a two-stage algorithm:

1) a packet pair technique to estimate the effective capacity of the wireless networks;
2) a packet train technique to estimate the achievable throughput and report the inferred available bandwidth.

In the first stage, n packet pairs are sent to estimate the capacity C_e. To mitigate the effect of cross and contending traffic, C_e is calculated as the median instead of the mean of the n dispersion samples from the n packet pairs. In the second stage, a train of m probe packets are sent at rate C_e. The achievable throughput A_t is equal to the average dispersion rate of the train. WBest uses a two-stage algorithm. In the first stage, *n* packet pairs are sent to estimate the capacity C_e. To mitigate the effect of cross and contending traffic, C_e is calculated as the median instead of the mean of the *n* dispersion samples from the n packet pairs. The median is used as opposed to the mean in order to mitigate the impact of outliers in the arrival distribution. In such cases, the mean results in a lower capacity estimate than does the median, and would make the second stage packet train less effective at accurately determining the available bandwidth.

$$C_e = median\left(\frac{L}{T(i)}\right) \qquad 3.23$$

where L is the packet size and $T_{(i)}$ is the dispersion corresponding to the i^{th} packet pair. In the second stage, a train of m probe packets is sent at rate C_e. The achievable throughput A_t is equal to the average dispersion rate of the train.

$$A_t = \frac{L}{mean(T_i, i=1,\ldots.m)} \quad 3.24$$

The advantage of WBest is that it does not depend upon search algorithms to detect the available bandwidth but instead, statistically detects the available fraction of the effective capacity, mitigating estimation delay and the impact of random wireless channel errors.

All the active techniques cited above present various drawbacks in as far as mobile ad hoc networks are concerned. First, when many nodes in an ad hoc network need to perform such an evaluation for several destinations, the number of probe packets introduced in the network can be important and interact with the traffic and with other probes. Secondly, a mobile network can contain links of heterogeneous quality. An end-to-end evaluation technique may not be as reactive as a local technique complemented with an appropriate measurements combination technique. Also, since probes are an active measure, the probes may not be able to determine an accurate value if packet loss occurs. Losses thus reduce the quality of the measurement. Also, because probing attempts to measure the medium access delay, priority queuing and priority medium access are required. Without these priorities, probe messages may incur lengthy queuing delays that distort the measured value.

Other measurement based method is Cprobe is a pioneering tool for estimating the available bandwidth using end-to-end measurements. Cprobe does not assume fair queueing. Instead of using a pair of packets, cprobe sends a short train of ICMP packets and computes the available bandwidth as the probe traffic divided by the interval between the arrival of the last ICMP ECHO and the first ICMP ECHO in the train.

Bprobe uses packet pair dispersion to estimate the capacity of a path. The original tool uses SGI-specific utilities to obtain high-resolution timestamps and to set a high priority for the tool process. Bprobe processes packet pair measurements with an interesting “union and intersection filtering” technique, in an attempt to discard packet pair measurements affected by cross traffic. In addition, bprobe uses variable-sized probing packets to improve the accuracy of the tool when cross traffic packets are of a few fixed sizes (such as 40, 576, or 1500 bytes). Bprobe requires access only at the sender side of a path, because the target host (receiver) responds to the sender’s ICMP-echo packets with ICMP-echo replies. Unfortunately, ICMP replies are sometimes rate-limited to avoid denial-of-service attacks, negatively influencing measurement accuracy.

Sprobe is a lightweight capacity estimation tool that provides a quick capacity estimate. The tool runs only at the source of the path. To measure the capacity of the forward path from the source to a remote host, sprobe sends a few packet pairs (normally TCP SYN packets) to the remote host. The remote host replies with TCP RST packets, allowing the sender to estimate the packet pair dispersion in the forward path. If the remote hosts runs a web or gnutella server, the tool can estimate the capacity in the reverse path – from the remote host to the source – by initiating a short file transfer from the remote host and analyzing the dispersion of the packet pairs that TCP sends during slow start.

3.3.2.10 Summary

Path and bandwidth estimation, as well as the previous methods described, all help to analyse and discover large amounts of data concerning networks. However, it must be considered whether these tools are sufficient in a world where the number of wired networks is decreasing, giving way to wireless networks. While many of the programs previously described might work without modification, others do not port so well. Table 9 provides a brief summary of path and bandwidth analysis tools.

3.3.3 Calculation-Based Approaches

Here, available bandwidth is inferred based on calculations taking into account several parameters such as estimating the channel capacity of a link using metrics like channel busy ratio. The channel busy ratio, is a measure of how much the channel is being utilized. In IEEE 802.11 based wireless networks, carrier sensing enables nodes to detect whether other nodes are in transmitting, receiving or sensing packet transmission. In (Chakeres & Belding-Royer, 2004) busy time is defined to be the total time within a time interval that a node is either transmitting packets, receiving packets or sensing packet transmissions.

Table 9: Summary of Path / Bandwidth Analysis Tools

Tool	**Author**	**Measurement metric**	**Methodology**
BWest	Li, Claypool,	Effective link capacity Achievable throughput	Packet pairs Packet trains
DietTopp	Johnsson, Melander,	Available bandwidth	Packet trains
TOPP	Melander	Available bandwidth	Packet pairs
Pathchar	Jacobson	Per-hop Capacity	Variable Packet Size
Pchar	Mah	Per-hop Capacity	Variable Packet Size
Bprobe	Carter	End-to-End Capacity	Packet Pairs
Pathrate	Dovrolis, Prasad	End-to-End Capacity	Packet Pairs & Trains
Sprobe	Saroiu	End-to-End Capacity	Packet Pairs
Cprobe	Carter	End-to-EndAvailable-bw	Packet Trains
Pathload	Jain, Dovrolis	End-to-End Available-bw	Self-Loading Periodic Streams
pathChirp	Ribeiro	End-to-End Available-bw	Self-Loading Packet Chirps

Given the network utilization *U* and the maximum bandwidth (B_{max}), the available bandwidth given by:

$$B_{avail} = (1-U)*B_{max} \qquad 3.25$$

To understand the busy time ratio, it is important to understand the behaviour of IEEE 802.11 networks. For nodes in space, we refer to the maximum separation between a sender and receiver for successful packet reception as the transmission range (*RxR)* as shown in Figure 28. Neighbours (N) are all nodes within the transmission range of a particular sender. Nodes that are within carrier sensing range of a sender can sense packet transmissions. Carrier sensing neighbours (CSN) are all the nodes within a sender's carrier sensing range (CSR). The carrier sensing range is the maximum distance that a node can detect an ongoing packet from another node. This range is typically much larger than the transmission range.

For a node to correctly receive packets there should not be other nodes in the CSR transmitting at the same time, otherwise there would be interference even if the senders are outside each other's CSR. The distance between a receiving node and another sender, such that this receiver's ability to decode a packet from its sender is not affected is called the receiver interference distance (RID). For example, in Figure 28, node X can transmit at the same time as node S without affecting packets received by node R from node S provided that node X is outside node R's RID, otherwise R is unable to successfully receive packets from node S. Nodes X and S are both outside each other's CSR so they do not prohibit each other from transmitting. The size of the RID depends on transmission power, minimum reception power, propagation model and hardware capture abilities.

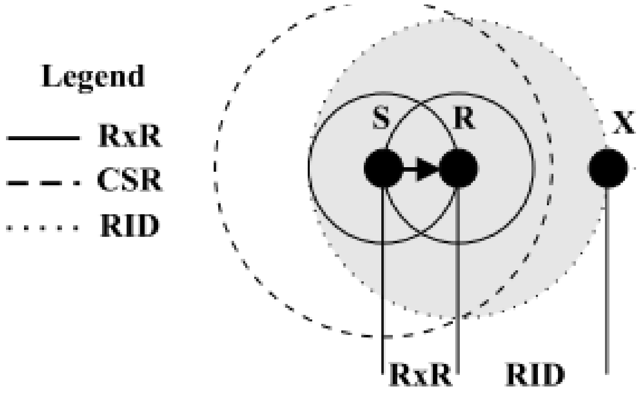


Figure 28: The interference distance between a receiver (R) and sender (X)

3.3.3.1 "Listening" Bandwidth Estimation using NAV

To estimate the available bandwidth, intuitively, a node listens to the channel to determine the channel state. In so doing, it is able to determine the available bandwidth it has every second. For a specific duration the channel can only be free when all nodes are idle, otherwise the channel is regarded to be busy as shown in Figure 29. An IEEE802.11 based node uses both a physical carrier sense and the Network Allocation Vector (NAV) to determine the free and busy times.

The Network Allocation Vector contains the time for which the network is allocated to be used by a node. A node wishing to transmit data, first senses the medium using virtual carrier sensing mechanism. Virtual carrier sensing considers the medium as idle if Network Allocation Vector (NAV) is zero, otherwise it considers the medium busy. Send state and receive state denotes the send and receive status of the node in question respectively. The calculation of the available bandwidth by each node in the network is based on the two timers in each node in the mobile ad hoc network.

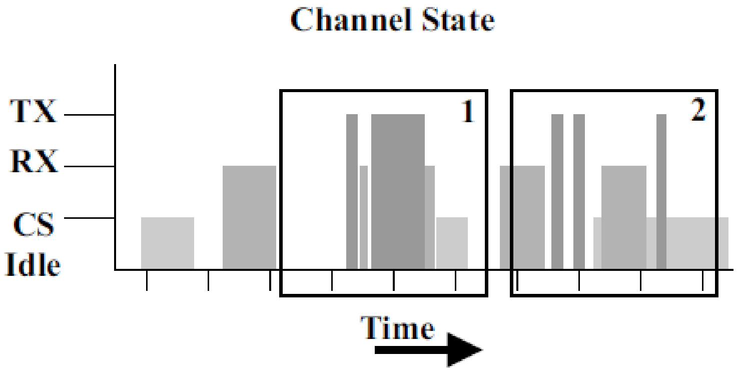


Figure 29: An example of channel state at different times

Each node in the network has two timers that is the idle timer and the busy timer. If the node senses that the network is busy it increments the busy timer and freezes the idle timer and vice-versa. The overall time equals the total idle time and busy time. The MAC detects that the channel is busy when NAV's value is greater than zero, receive state changes from idle to any other state, or send state changes from idle to any other state. If any one of these conditions is satisfied the network will be busy and hence it increments the busy timer and freezes the idle timer.

The available bandwidth for new data transmissions is the ratio of free time to overall time multiplied by the channel bandwidth, divided by a weight factor. The weight factor is introduced to carter for the overhead introduced by the DIFS, SIFS and backoff scheme of the IEEE802.11 MAC layer. This overhead makes it impossible in a distributed MAC competition scheme to fully use the available bandwidth for data transmission. The "Listen" bandwidth estimation formula for calculating the residual bandwidth is:

$$A_{bw} = \frac{C\left(\frac{F_{time}}{O_{time}}\right)}{W} \qquad 3.26$$

Here, A_{bw} is available bandwidth, C is channel bandwidth F_{time} is free time and O_{time} is overall time and W is a weight factor.

The "Listening" bandwidth estimation method is suitable in a stable network, which has low node mobility. However, in an unstable network with routes between any two nodes breaking, a node cannot release the bandwidth immediately because it does not know how much bandwidth each node in the broken route consumes.

3.3.3.2 Listening Bandwidth Estimation Algorithm

Each node in the network has two timers, which are the idle timer and the busy timer. Each node should sense every second whether the network is busy or idle. If the network is busy, the node will increment the time for which the network has been busy and freeze the idle timer and if the network is idle the node will increment the time for which the network has been idle and freeze the busy timer. In our algorithm design for the listening method, we are repeating the simulations for every 30 seconds.

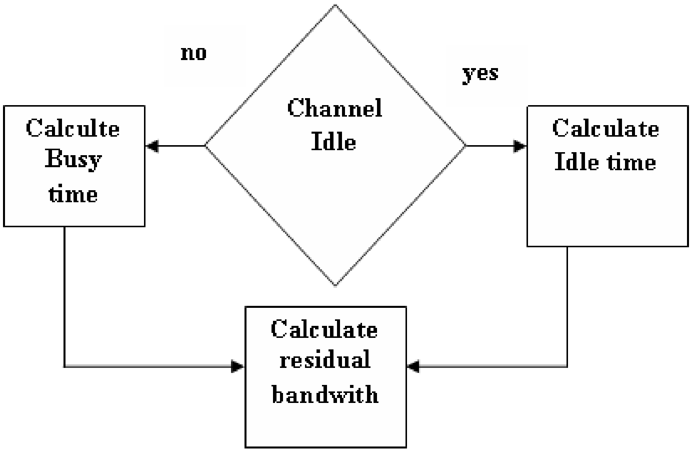


Figure 30: The listening bandwidth estimation mechanism

The total time will be 30 seconds and this is divided into the idle time and busy time. The node will be listening to the activities of its interference range. If within the node's interference range, there is some sending and receiving that means the network is busy. If there is no activity of sending and receiving within a node's interference range, then the network will be idle. The flow diagram for the Listening bandwidth estimation method is shown in Figure 30. Using the listening flow diagram we can easily formulate the listening bandwidth estimation algorithm as shown below.

Listening bandwidth estimation algorithm

```
Repeat
  i.    For i =1 to n
  ii.   Check if (NAV value <CurrentTime,
                  ReceiveState = Idle
                  and SendState = idle)
  iii.  Compute Idle Time
  iv.   Compute Available bandwidth using weight factor
  v.    Store available bandwidth
```

The Listening method’s weakness of low accuracy when a route is broken means that another method must be found to compliment it. Therefore, (Chen & Heinzelman, 2005) introduced another approach called “Hello” bandwidth estimation that is better able to reallocate available bandwidth when routes break.

3.3.3.3 “Hello” bandwidth estimation method

In the “Hello” bandwidth estimation method (Chen & Heinzelman, 2005), the source node’s current bandwidth usage and its one-hop neighbour’s current bandwidth usage is added onto the routing protocol “Hello” message. Each node will then use the information provided in the Hello messages and its knowledge of the frequency re-use pattern to estimate its available bandwidth. In this way there are no extra control messages to disseminate the bandwidth information.

In the IEEE 802.11 MAC, nodes are allowed to access the channel only when the channel is free. A node will detect that the channel is free if no nodes are transmitting packets within its interference range which is normally twice the transmission range. The frequency can therefore, be used by nodes outside of the second neighbouring node’s range. The bandwidth in the two-hop circle varies with the topology and the traffic status, but the raw channel bandwidth is the soft upper bound bandwidth in the estimation to approximate the bandwidth usage. With the above frequency reuse pattern, we can estimate the bandwidth within the interference range of a node. In this way a node can estimate its available bandwidth based on information from within its interference range.

The first neighbouring hosts’ information can be obtained directly, but there is no way to get the second neighbouring hosts’ bandwidth information directly. One way of getting such information is by disseminating the host bandwidth information using higher transmission power to reach the two-hop neighbourhood. However, this is bound to consume more power, which is a scarce resource in MANETs. Using more also destroys the frequency reuse pattern and introduces more interference. In (Chen & Heinzelman, 2005) they proposed using the information in the Hello messages received from neighbours to learn about bandwidth consumed by the second neighbouring nodes. AODV uses the "Hello" messages to update the neighbour caches.

Once a host receives a "Hello" message from its neighbours, it determines whether this "Hello" is an updated one by examining the message's time stamp. Once a host knows the bandwidth consumption of its first neighbours and its second neighbours, the available bandwidth estimation becomes simple. The residual bandwidth is simply the raw channel bandwidth minus the overall consumed bandwidth, divided by a weight factor.

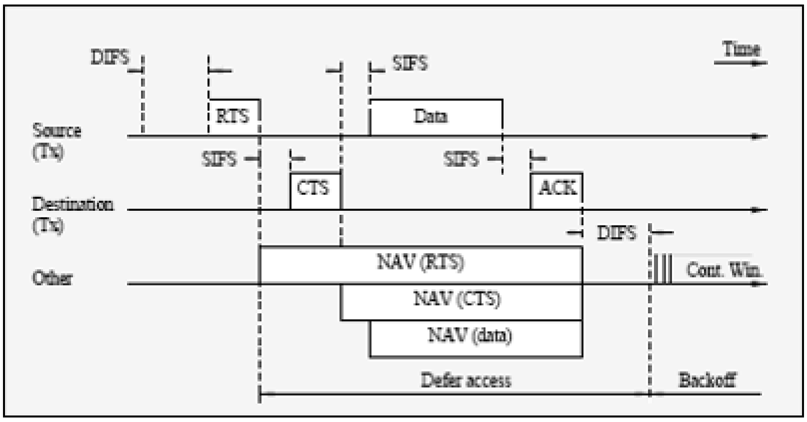


Figure 31: RTS/CTS Access Scheme

We need to divide the residual bandwidth by a weight factor due to the IEEE 802.11 MAC's nature and the overhead required by the routing protocol. The relationships of RTS, CTS, ACK and some inter-frame spacing can be shown in Figure 31. The hello message residual bandwidth estimation is given by:

$$A_{bw} = \frac{C - B_{cons}}{W} \qquad 3.27$$

Here, B_{cons} is the overall consumed bandwidth and C is the channel capacity and W is the weight factor.

3.3.3.4 Hello Bandwidth Estimation Algorithm

The Hello bandwidth estimation method is basically used in case of the network break up. This is due to the failure of the Listening method to reallocate the available bandwidth of the node(s) which will have moved and hence give poor bandwidth approximates, when the network break (normally caused by the movement of a node(s) outside the transmission range of a node in question). When the estimates are more than real, this normally would cause congestion of the network, dropping down of packets and waste of the scarce battery power of the mobile ad hoc nodes. Underestimated available bandwidth value leads to the underutilisation of the scarce network resource which is the bandwidth. Before using the Hello method, the node in question should check whether the route is broken or not. If the route is not broken the node should estimate the available bandwidth using the Listening method otherwise the node should use the Hello message method. In case of a broken route, the node should then gather the total bandwidth which is been consumed by its first neighbours and its second neighbours. The bandwidth consumed by the first neighbours of the node in question can be found directly since these neighbours can communicate with the node in question directly. But it is difficult to get the second neighbours' consumed bandwidth directly.

In (Chen & Heinzelman, 2005) they used AODV "Hello" messages to update the neighbour caches. The normal AODV "Hello" message only keeps the address of the node which initiates the message. They modified the "Hello" message to include two fields, one including (host address, consumed bandwidth, timestamp), and the second field including (neighbour's addresses, consumed bandwidth, timestamp) as shown in Table 10. The time stamp is used by a node to determine whether the "Hello" message, received from its neighbour, is an updated one by observing the message's time stamp. They used the cache structure shown in Figure 32, which includes a first neighbour table and a second neighbour table. The second neighbours are linked with their corresponding first neighbours in cache.

Table 10: Modified Hello Structure.

ID	**Consumed bandwidth**	**Timestamp**
Neighbour ID 1	Consumed Bandwidth	Timestamp
.	.	.
.	.	.
Neighbour ID n	Consumed Bandwidth	Timestamp

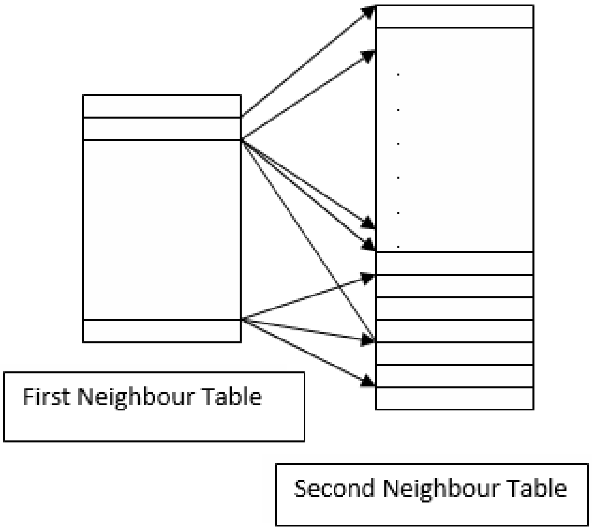


Figure 32: Neighbour cache structure

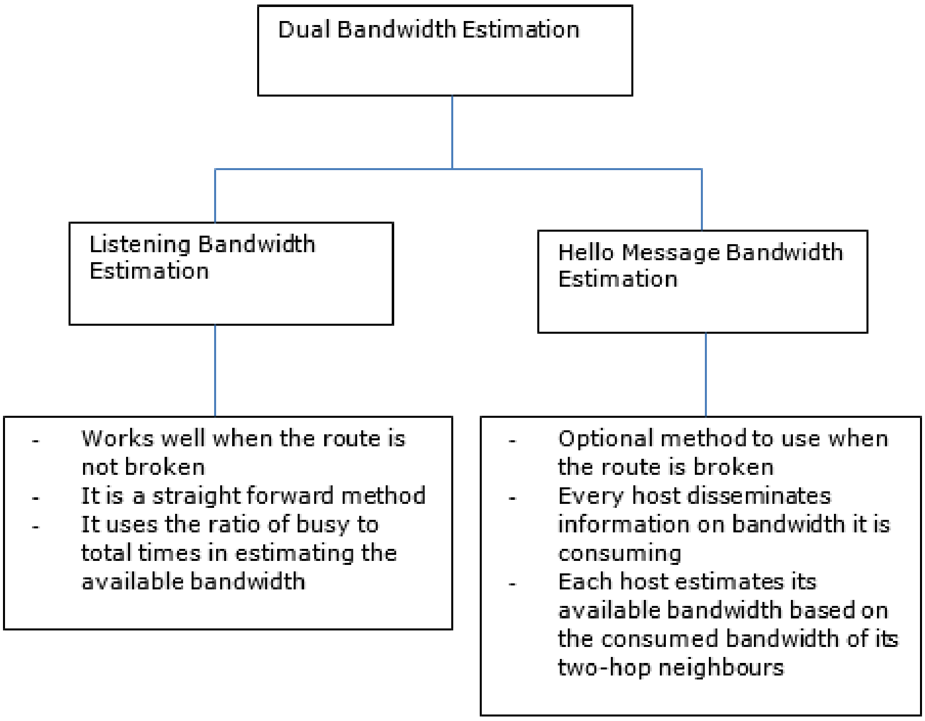
Using the flow diagram for the Hello message bandwidth estimation above, we can formulate the Hello bandwidth estimation algorithm as shown below.

Hello Bandwidth Estimation Algorithm

```
/* n is the number of nodes*/
          For (i =1 to n)
    i.    Create a First hop Neighbour Cache
   ii.    Add bandwidth values of all first hop neighbours (1$^{st}$HopBandwidthTotal)
  iii.    Create a second Neighbour Cache
   iv.    Add bandwidth values for all second hop neighbours
          (2ndHopBandwidthTotal)
    v.    Total Bandwidth = 1stHopBandwidthTotal + 2ndHopBandwidthTotal;
   vi.    Calculate available bandwidth using weight factor
  vii.    Store Available Bandwidth value
 viii.    Delete a node(s) which fail to send the message and add new node(s)
          which have entered the node's transmission range*/
```

3.4. Model Formulation

In this section, we design and use the dual bandwidth estimation method (DBE) to calculate the residual bandwidth before each node does the transmission. The Dual bandwidth estimation method is made up of the two methods, which are Listening bandwidth estimation and Hello messages bandwidth estimation as shown in Figure 33. The figure shows also the best scenarios in which each of the methods can best estimate the available bandwidth.


Dual Bandwidth Estimation

Listening Bandwidth Estimation

- Works well when the route is not broken
- It is a straight forward method
- It uses the ratio of busy to total times in estimating the available bandwidth

Hello Message Bandwidth Estimation

- Optional method to use when the route is broken
- Every host disseminates information on bandwidth it is consuming
- Each host estimates its available bandwidth based on the consumed bandwidth of its two-hop neighbours

Figure 33: The structure of the Dual bandwidth estimation method

The DBE method will be coupled with the traffic different scheme to come up with a QoS –aware MAC layer, which would allow any packet to access the wireless media based on the priorities and also the network available bandwidth. The main aim of this research is to propose an adaptive traffic differentiation scheme which admits traffic based on the network available bandwidth, so as to reduce the congestion rate, improve channel utilization and also reduce energy dissipation. The flow diagram for the Dual bandwidth estimation is shown in Figure 34.

Before transmitting any data packet, each node in the network should check whether its existing network is broken or not. If the network is broken the node should then estimate the available bandwidth using the Hello messages bandwidth estimation since the method can reallocate the available bandwidth in the case of breakup of the network.

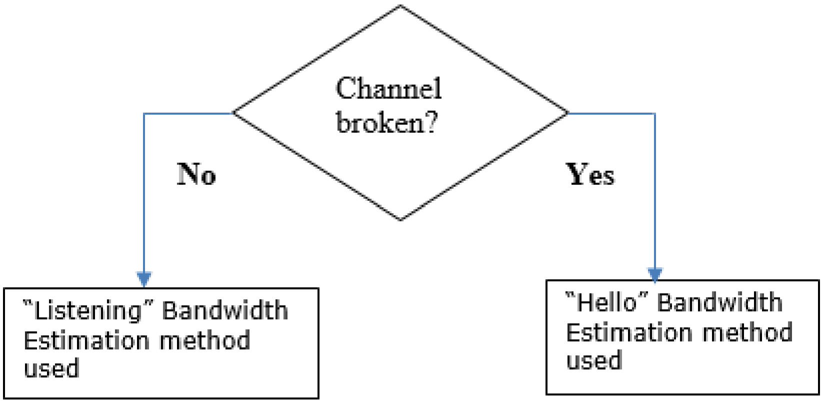

Channel broken?
No
Yes
"Listening" Bandwidth Estimation method used
"Hello" Bandwidth Estimation method used

Figure 34: The flow diagram of the Dual Bandwidth Estimation method

Hence, offering better estimates than listening bandwidth estimation in this case. When the network is not broken or stable, each node should estimate the available bandwidth using the listening bandwidth estimation method. The more stable the network is, the more often it will resort to use the listening bandwidth estimation method and then vice–versa for the Hello messages bandwidth estimation. Although the new algorithm is more complex than each of the individual algorithms (listening and hello methods), the advantage of getting an accurate and reliable bandwidth estimation outweighs the disadvantage of using two methods.

3.5. Experimentation and analysis

Simulation for the proposed algorithms has been carried out in C++ code. The AODV routing protocol and modified AODV (MAODV) are analysed with respect to proposed bandwidth estimation by taking different configurations and scenarios.

Figure 35 shows the model of system simulated. We consider node x surrounded by nodes y within the second hop neighbourhood of *x*. The traffic from nodes *z* does not affect the available bandwidth of node *x* since these nodes are outside node *x*'s interference range R.

For the listening bandwidth estimation method, the state of each of nodes *y* was determined from a data set we created. When the state have data to send or receive, then it is busy otherwise it is idle. These states are used over time to calculate the total idle time and subsequently the available bandwidth for node *x*. The network will by default use listening bandwidth estimation method unless the network is unstable and the path has been broken. The change of the network state from stability to instability is represented as the change of the generated random numbers 0 or 1.

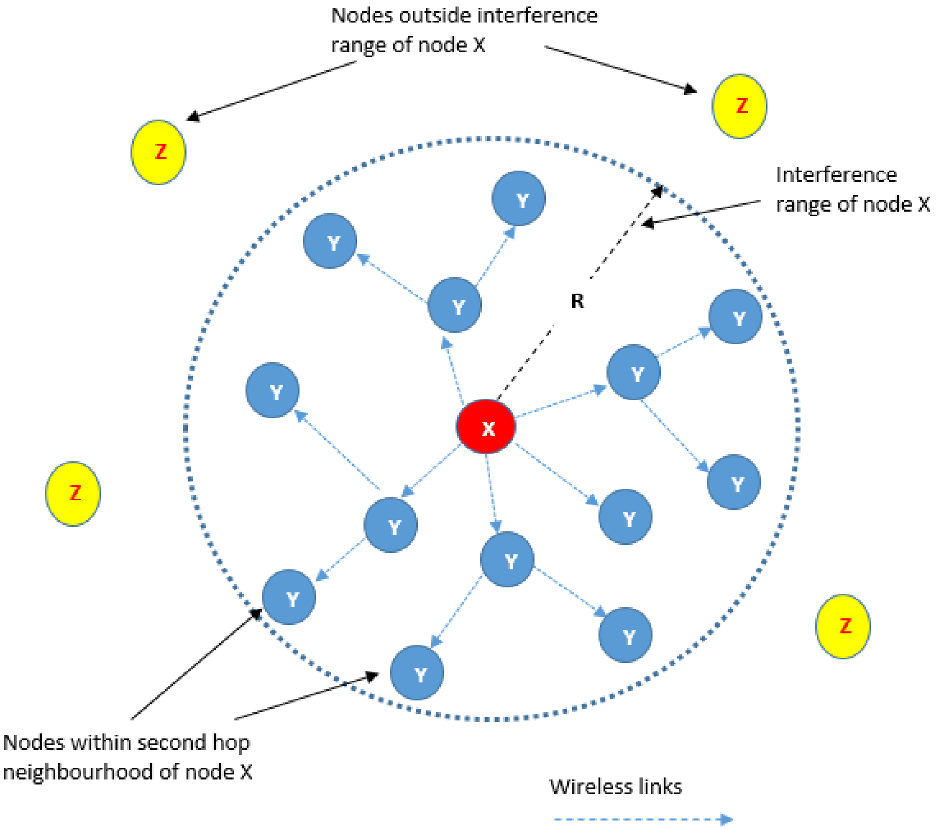

Nodes outside interference range of node X
Z
Z
Interference range of node X
R
X
Y
Z
Z
Nodes within second hop neighbourhood of node X
Wireless links

Figure 35: The simulated environment for available bandwidth of node *x*

To determine the network state, a random number (0 or 1) is generated, such that a 1 indicated that the path was broken and a 0 means that the network is stable. If the network is stable, the DBE method will resort to use the Listening method and in case of network break up the bandwidth calculation process will switch to use the Hello method, which can reallocate the bandwidth consumed by the node within the broken routes and hence offering better estimates.

A dataset was created with the information of amount of data sent by each node every time. Data sent by nodes *y* per second is added to come up with consumed bandwidth. This is subtracted from the channel capacity to give available bandwidth. Depending upon the network conditions the DBE method is used. The listening method is used to calculate the bandwidth, if the network is stable. Hello method is used to calculate the bandwidth in case of break up in the network.

3.6. Results

Table 11: The simulation results of the DBE for the first 10 seconds

Breaking rate/s	Use of Hello method	Use Listen Method	Packet loss rate/s
0	0	401	0
5.21	63	338	9.8
5.67	67	334	10.66
6.84	74	327	13.14
8.25	79	322	15.84
8.3	80	321	16.06
8.6	87	314	17.57
8.62	87	314	17.43
11.72	123	278	21.73
12.24	123	278	22.69
12.25	123	278	22.72
13.87	135	266	25.72
18.09	188	213	34.07
20.17	188	213	37.98

After running several simulations, results were summarized and analysed. Part of the results are as shown Table 11**,** which shows results after many simulations were run for a period of 10 seconds each time. The network breaking rate changed and each time the number of lost packets was recorded. The number of times the hello method was used was also recorded after every simulation.

From the simulations we have found out that as the breaking rate increases, the use of the listening method decreases and the use of "hello" message increases as shown in Figure 36. This shows that the dual bandwidth estimation method performed as planned. The system used the listening as the default method and when the network broke, it resorted to the "hello" method.

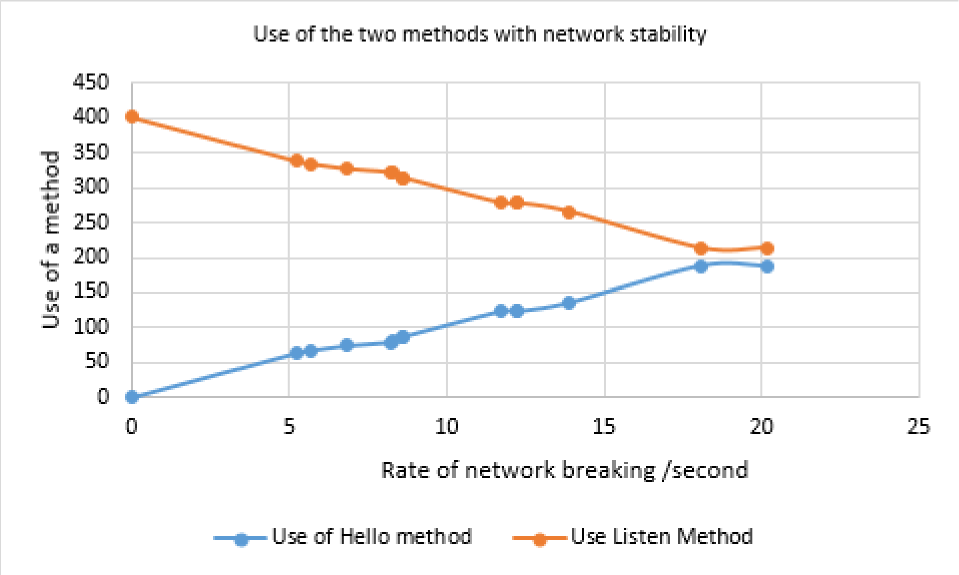


Figure 36: Usage of the two methods in relation to network stability

Network breaking is caused by node mobility when some nodes are moving outside the interference range whilst some are joining the network. Thus, there will be a great risk for the network to break when there is a lot of mobility and hence less likely use of the listening method to calculate the available bandwidth.

Table 12 and Figure 37 shows the variation of packet loss rate with the increase in network instability. The packet loss rate have a variance of 3 packets per second. The packet loss rate increases when the rate at which the network was breaking. The reason is that when path is broken, packets will be dropped until a new path is found.

Therefore the relationship between packet loss and the rate at which the network is breaking is nearly linear. The dispersion of packet loss values is quite huge meaning that the values were much spread from the mean value each time. This is because in some periods there was very low packet loss and some periods had pronounced packet loss. In some instances there was network breaking but no recorded packet loss. This may be attributed to situations where no packets were sent through that broken link but sent through a new found route.

Table 12: The variation of packet loss with the network-breaking rate

Average breaking rate	Average packets lost	standard deviation
5.25	9.16	5.45
5.58	10.16	4.48
6.9	12.8	5.6
8.11	15.77	4.66
8.22	15.77	5.53
8.5	17	8.06
8.7	17.4	8.3
11.8	21.4	8
12.1	22.6	6.69
12.11	22.8	8.77
13.67	25.33	8.27
18.19	34.2	13.75
18.33	38	18.71

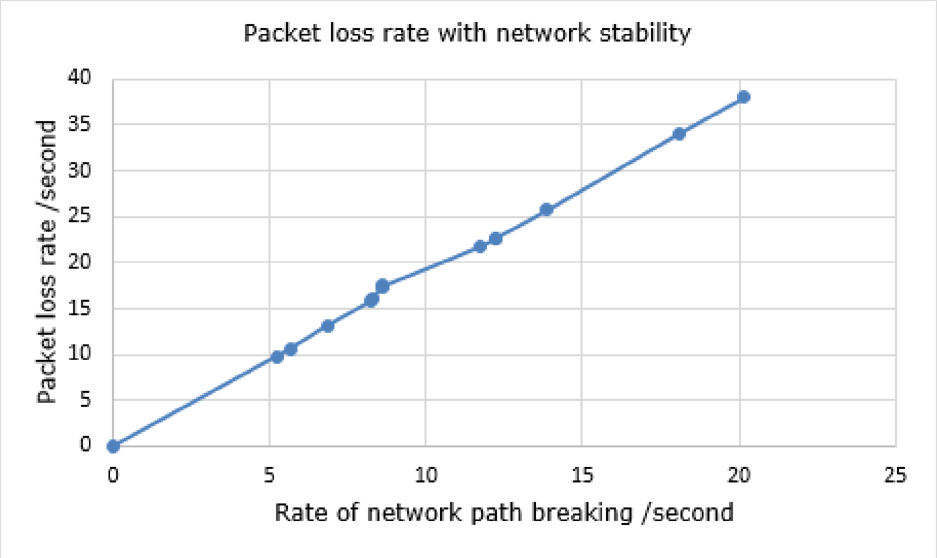


Figure 37: Effect of network path breaking on pcket loss

Table 13: Variation of available bandwidth with the number of nodes

Number of nodes	Average bandwidth KB/s	Standard deviation KB/s
3	26.53	0.69
5	24.2	0.93
8	20.74	1.14
10	18.45	1.23
12	16.16	1.52
15	12.97	1.55
20	7.97	1.61
23	4.97	1.96
25	3.07	2.14
30	0.67	1.16
40	0	0

Table 13 and Figure 38 show the variation of average available bandwidth with the number of in the network. The network stability constant and varied the number of nodes. It is evident that the available bandwidth decreases as the number of bandwidth consuming nodes increases. However the dispersion of the bandwidth values measured increased with the number of nodes. Available bandwidth is almost inversely proportional to the number of nodes.

Therefore the relationship between packet loss and the rate at which the network is breaking is nearly linear. The dispersion of packet loss values is quite huge meaning that the values were much spread from the mean value each time.

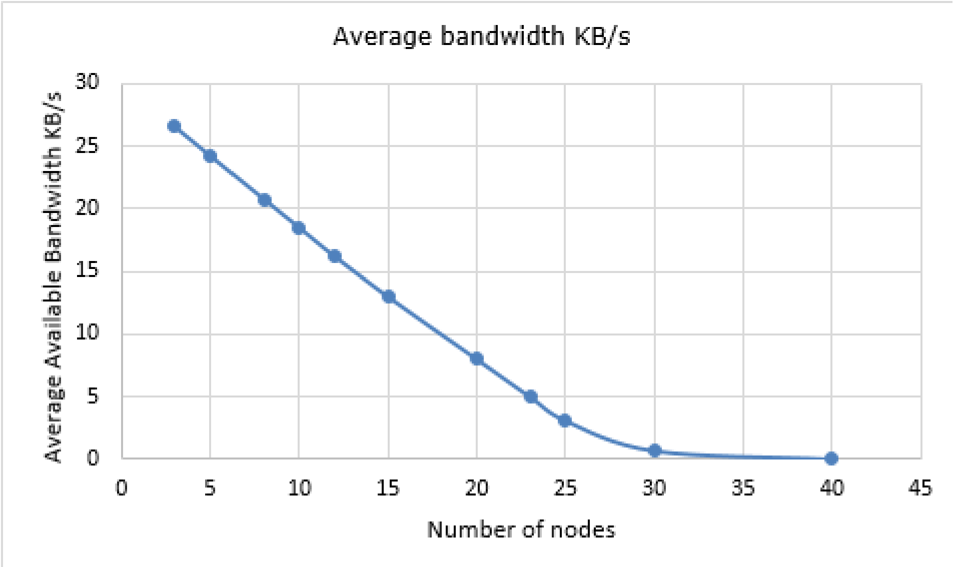


Figure 38: Variation of available bandwidth with the number of nodes

Table 14: Variation of throughput against number of nodes in the network

Number of nodes	Average Throughput KB/s	Standard Deviation KB/s
3	664.67	2.58
5	1132.61	2.97
8	1829.67	1.66
10	2291.17	1.79
12	2751.94	2.78
15	3390.56	2.04
20	4396.06	1.23
23	4998.17	1.86
25	5383.61	1.72
30	5862.39	2.58

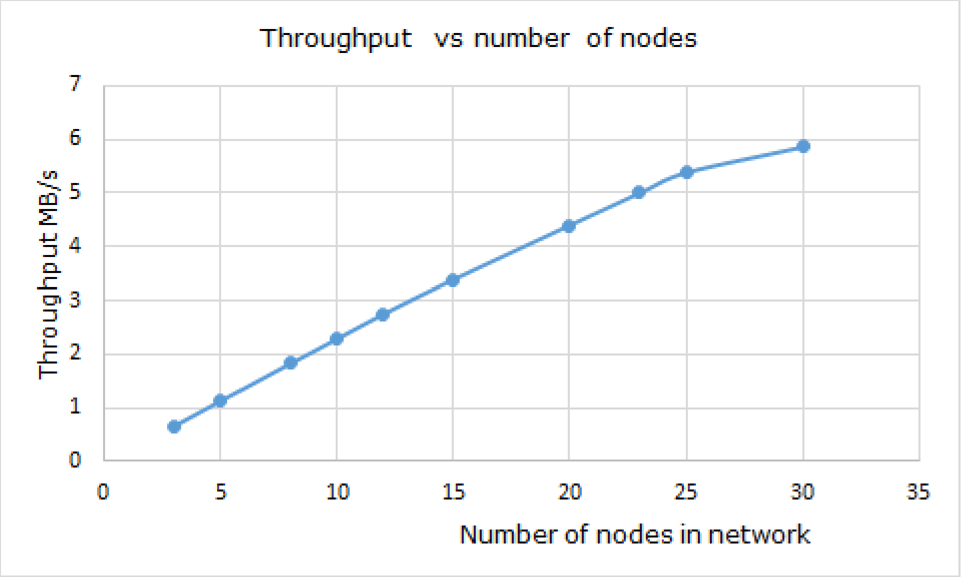


Figure 39: Variation of throughput with the number of nodes in the network

The wireless media has a limited capacity and when that capacity is reached then throughput will level out. In this case we had put the network capacity at 6MBps. We can see that the graph of throughput is the direct inverse of available bandwidth since available bandwidth is the difference of used bandwidth (throughput) from the channel capacity.

3.7. Conclusion

In this chapter we studied various bandwidth estimation methods. This included measurement based, model based and calculation based bandwidth estimation methods. We concluded that calculation based estimation methods are most suitable for MANET environment since they are not computationally intensive and non-intrusive. We looked at two calculation based methods, listening and hello methods proposed in literature. The listening method utilises the NAV and the RTS/CTS messages to identify busy and idle times in the network. The hello method modifies the routing protocol hello messages to carry bandwidth utilization information. We went ahead and proposed a bandwidth estimation method which encompasses these two methods, which we called Dual bandwidth estimation method.

Our results showed that the Dual Bandwidth Estimation method works as good as than both the listening bandwidth estimation and the hello messages bandwidth estimation methods, without the weaknesses associated with either of the methods. This is because it combines the strengths of the two methods and avoids their weaknesses. It is an effective method in the case of the network instability since it can offer better estimates if the network breaks up. It also maintains the simplicity and robustness of the listening method in such a way that bandwidth estimates are valid all the time notwithstanding the time between the routing protocol's hello packets. Ad hoc networks provide a method for communicating between mobile devices without requiring an infrastructure. Direct communication is possible between nodes that are located within range and intermediate nodes are used to route messages to destinations beyond a single hop. Bandwidth utilization is a big problem in real time networks. This shows that the Dual bandwidth estimation method is a good tool when we study bandwidth management as investigated in Chapter 4.

4. Bandwidth Management Framework for MANETs[3]

4.1 Problem definition

For a network to be able to provide QoS all the layers in OSI model should cooperate to come up with a sufficiently acceptable performance of the network. The transmission quality is taken care of by the physical layer. For example by the physical layer can adaptively increase or decrease the transmission power depending on the availability of resources. In the same way, the link layer should react to the changes in the link error rate, for example by including the use of automatic repeat-request technique. The routes with sufficient resources are found by QoS-Routing and QoS-Signalling to allocate and release bandwidth depending on the network conditions.

Our endeavour to provide QoS to multimedia traffic calls for a QoS-adaptation that hides all the environment related features from the awareness of multimedia applications. QoS adaptation provides an interface for applications to submit their requirements and is responsible to dynamically react to QoS changes for a certain flow, according to these requirements. This chapter concentrates on QoS-Signalling and adaptation and less on the MAC layer and Physical layer. We require a framework that will allow the network to receive QoS level requirements from a new flow, perform QoS admission procedure, allocate resources and schedule to the flow. The framework should ensure that, if resources are no longer present, for one reason or the other, the network should be able to institute QoS adaptation to self-heal and revert to acceptable

[3] The work in this chapter is an extension of work published in the following conference proceedings article; Nyambo B.M., G.K. Janssens and W. Lamotte (2015). Adaptive Mobile Ad Hoc Networks (AMAN): a QoS framework for mobile ad-hoc networks. In: P.J. Sequeria Gonçalves (ed.), *Proceedings of the 21st European Concurrent Engineering Conference/11th Future Business Technology Conference/19th Annual Euromedia Conference (ECEC'2015 / FUBUTEC'2015 / EUROMEDIA'2015),* Lisbon, Portugal, 27-29 April 2015, pp. 42-48 (ISBN 978-90-77381-88-5).

levels. We adopt some principles of signalling, reservation and adaptation from INSIGNIA and ASAP to come up with a new framework that operates effectively under low bandwidth environments but maintaining acceptable QoS levels for multimedia and high priority traffic.

The main aim of this chapter seeks to come up with a framework that focuses on optimizing the bandwidth allocation and adaptation strategies in MANETs without waste of these resources and ensuring service performance in MANETs carrying multimedia traffic is enhanced whilst guaranteeing Quality of Service (QoS) and addressing the major challenges of MANETs outlined previously. Some schemes address Resource Reservation and Admission Control but they have some limitations in guaranteeing QoS in MANETs especially for multimedia applications, which require greater bandwidth allocations. Thus, the main thrust of the research focus on a scheme that also relies on bandwidth estimation and allocation of the available bandwidth since most of the existing schemes are mainly based on service differentiation only, which provides prioritization of service classes without giving hard guarantees. Other QoS Schemes like INSIGNIA, SWAN and ASAP differentiate traffic into two classes, namely real-time traffic and best-effort traffic. In this research we propose to differentiate traffic into multiple classes depending on the use and importance of the data.

4.2 System Definition

In a general MANET, if a new flow request admission, it is allowed to contend for bandwidth with ongoing flows. In so doing this new flow will interfere with ongoing flows and may cause QoS degradation on flows. The new flow may also be admitted but the resources would be too few for successful transmission. We need to design a new QoS scheme such that the network becomes intelligent enough to a scenario of admission but no transmission. From the literature (Abbas & Kure, 2008), (Khalfallah, Sarr, & Guerin Lassous, 2007), (Oh, Marfia, & Gerla, 2010) (Yu, Navaratnam, & Moessner, 2013),we can conclude that, the implementation of the proposed traffic management scheme is supposed to fulfil quality of service requirements for multimedia traffic:

i. Admission of a new flow into the network only if enough bandwidth is available to carry the flow without interfering with other ongoing traffic. We don’t want the new flow to be admitted but for transmission to fail.

ii. Increase in the available bandwidth by reducing the allocation of other ongoing applications in order to incorporate a new flow. When the requested bandwidth of a new flow is greater than the available bandwidth then the other applications can reduce their allocations to the required minimum and the released bandwidth is added to the available bandwidth for use by the new flow. This takes place in the Bandwidth Adaptation Module.

iii. Denial of a new flow, of which the requested bandwidth is larger than the available bandwidth after Bandwidth Adaptation has been done. This also is done in the Reservation Module.

iv. All this have to be achieved through resource estimation, signalling, admission control and bandwidth adaptation

4.2.1 Resource Estimation

It is very essential to have an accurate estimation of available resources on network links or on end-to-end paths for many functions in networking such as admission control, load balancing, QoS routing, congestion control etc. Bandwidth is a fundamental resource so bandwidth estimation is very important in mobile wireless networks since the bandwidth of these networks is limited and is ever-changing. The bandwidth of a path is shared by the traffic under consideration and other traffic following in the neighbouring nodes. This reduces the amount of bandwidth available to the hosts. The other traffic is referred to as cross traffic. Available bandwidth is the amount of bandwidth “left over'' after the cross traffic. The link with the lowest available bandwidth is not necessarily the link with the lowest capacity. In this chapter the dual bandwidth estimation method described in chapter 3 is part of the bandwidth management framework.

4.2.2 Signalling

A MANET signalling system is supposed to consume very low bandwidth and it should be able to react fast enough to network dynamics on time-scales close to call and transmission speeds. If there are topology changes, the signalling system should be highly responsive to flow re-routing by re-establishing active reservations along the new path with little or no disruption to on-going flows. There are basically two signalling methods available for MANETs: in-band and out-band signalling systems. In-band signalling in its system, which means that the control information is carried alongside data unlike, out-of-band signalling systems where the control information is typically carried in separate control packets and on channels that may be distinct from the data path. In-band signalling is lightweight since control signals do not consume extra bandwidth by contending with traffic data since control information is included in the same data packet.

In-band signalling systems like INSIGNIA and ASAP can restore the flow state (i.e., a reservation) in response to topology changes within the interval of two consecutive IP packets under ideal conditions. In in-band signalling, performance relies on the speed at which the routing protocol can recompute new routes if no alternative route is cached after topology changes. Out-of-band signalling systems, for example, need to maintain source route information and respond to topology changes by directly signalling intermediate routers on an old path to allocate/free radio resources. In many cases, this is impossible to do if the affected router is out of radio contact from the signalling entity that attempts to deallocate resources over the old path.

In order to improve the working of the QoS framework, admission control can be coupled to the routing protocol. The coupling can either be loosely coupled, closely coupled or de-coupled. In the de-coupled option, the signalling and the routing protocols work independently of each other. This means that periodic network monitoring messages have to be sent to detect any topology changes. In the loosely coupled option, the signalling and routing interact with each other in a bi-directional manner. The signalling may provide feedback information to the routing layer regarding the route chosen and ask the routing protocol to for alternate routes if the chosen route fails to satisfy the required QoS levels. In the closely coupled approach the routing and signalling information are embedded in the same packet. QoS routing tries to find routes satisfying the minimum QoS requirements.

4.2.3 Admission Control

When a mobile node wants to initiate a new flow, it has to investigate available resources on the path towards the destination node before admitting the flow. The network should investigate and provide a path, from source to destination, containing enough free resources to carry a flow, without interfering with nearby ongoing traffic. Every node in the path from the source to destination should be able to decide whether or not to accept the new flow after analysing the available resources and the traffic already admitted.

The new call is admitted only if the traffic rate, combined with corresponding interferences, is smaller than the minimum of the available bandwidth of each node belonging to the path, and their respective first hop neighbours. Thus, the decision whether sufficient resources are available for a new flow or an aggregate of flows at the requested QoS without violating the existing QoS commitments to other applications depends on resource management policies and resource availability. Resources can only be reserved once the admission testing has been successfully completed and then committed later if the end-to-end admission control test is successful. Admission control is invoked by the resource reservation protocol before reservation is executed.

4.2.4 Resource Reservation

When there are enough resources to carry a flow without interfering with other on-going traffic then resources can be allocated to a new flow. If reservation is done by a two-pass mechanism, in the first pass data on network conditions is gathered and the second pass does the actual reservation. Two-pass reservation schemes avoid wasting resources but their drawback is their latency and this can be critical in a highly dynamic environment where the topology changes frequently and new routes have to be found now and again. One pass reservation uses one control message to do the actual reservation. This however leads to a waste of resources for some time since some bandwidth is allocated and never used. Reservation can be classified as soft reservation or hard reservation.

For hard reservation, a virtual circuit is established for the whole duration of the connection and the reserved resources are fixed for the specific transmission. When resources are hard reserved, the reservation record is always kept until an explicit release message is sent. The disadvantage of this is that it is not flexible enough for MANETs where the path and reservation need to dynamically respond to topology changes in a timely manner. Under soft reservation, reservation does not change node traffic characteristics. The reserved resource can still be used by Best Effort traffic and some other QoS flows, which temporarily need extra bandwidth. Hence this traffic is not affected by a soft reservation. Soft reservation increases the efficiency of resource utilisation both for QoS traffic and best effort. Soft sate reservations have a lifetime. After a certain time, the timer times out and the soft reservation and the timer have to be refreshed. Soft reservation is the most suitable approach for mobile ad hoc networks. Once a node loses connection, under soft reservation, there is no need of sending signals to change reservations but the timer would just timeout after sometime and the reservation is removed.

This research is part of efforts to come up with an optimised Resource Reservation scheme that addresses the challenges posed by MANETs and other issues like bandwidth estimation and adaptation, congestion control and admission control. This chapter therefore presents the scope of this research and proposes a scheme for traffic management in MANETs, clearly highlighting the part played by this research towards coming up with a comprehensive Resource Reservation Signalling scheme in MANETs. An overview of the proposed architecture is provided together with the relevant tools to be used.

4.2.5 Adaptation

The QoS situation in a MANET is very unpredictable. It can change rapidly and dramatically all the time due to the characteristics of the wireless link and mobility. When a link breaks and even when there are local repair mechanisms, the QoS cannot be guaranteed to be the same on the new path. Sometimes the new path will have less resources thereby creating a bottleneck. On the other hand if less traffic is in the network or when other traffic releases bandwidth, then the available bandwidth may increase. Therefore after reservation, the QoS framework must actively monitor the network dynamics and adapt the bandwidth allocation to flows according to some laid down strategy.

4.3 Model formulation

This section describes a framework architecture to realise the proposed QoS model. An overview of the proposed architecture is illustrated in Figure 40.

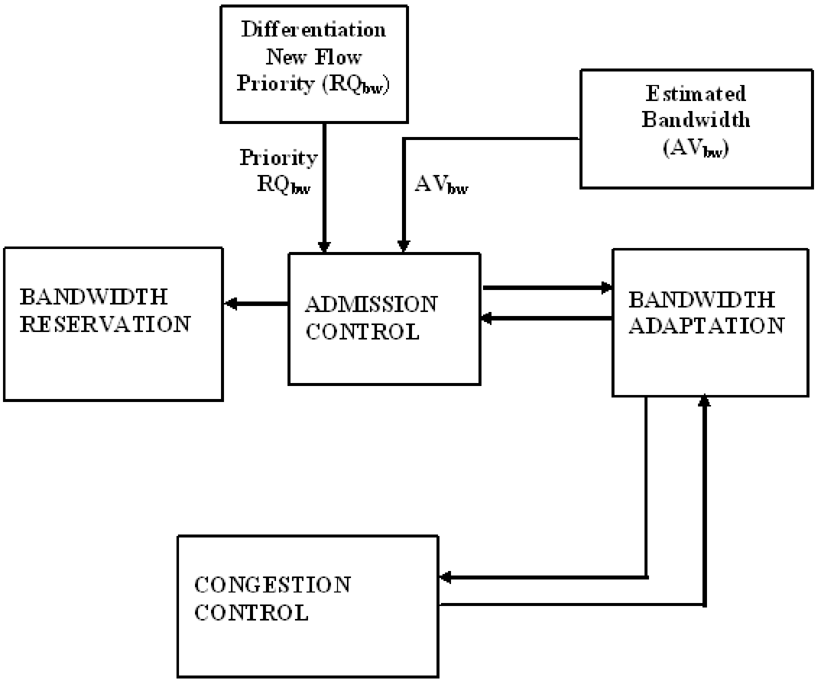


Figure 40: Traffic Management Framework Architecture

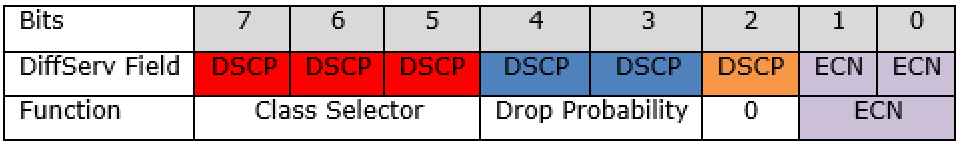
The proposed QoS architecture has six basic modules namely Bandwidth Estimation, Bandwidth Adaptation, Congestion Control, Admission Control and Reservation.

- Traffic differentiation – traffic have to be classified according to importance and bandwidth requirements. Real-time traffic like video and audio have to be given higher priority than other kinds of traffic.
- *Bandwidth Estimation* – Responsible for coming up with estimates of the available bandwidth within the network at any given moment.

- *Admission Control* – Responsible for comparing the resource requirements arising from the requested QoS against the available resources. This is invoked by the resource reservation protocol in the routing protocol before reservation is executed.
- *Reservation* – This module is closely coupled to the QoS Routing Protocol. The routing protocol is responsible for finding the path for a flow or aggregate of flows and maintaining the path at the required QoS level. Reservation will reserve bandwidth for each flow on the selected path.
- *Bandwidth Adaptation* – This module is responsible for making bandwidth allocation adjustments in case of insufficient resources. If a new flow requests admission in a network where bandwidth is deficient, the bandwidth adaptation module will adjust bandwidth allocations of ongoing flows downwards to allow new flows to be admitted.
- *Congestion Control* – It performs bandwidth adaptation on traffic so that the allocations to traffic flows is sufficient enough but congesting the network.

4.3.1 Traffic Differentiation

For traffic differentiation we adopt the DiffServ model, as described completely in RFC 2474, for classification and marking packets. DiffServ provides QoS by dividing traffic into a number of classes and allocating network resources on a per class basis. The class is marked directly on the packet in the 6 bit DiffServ Code Point (DSCP) field, which is part of the original type of service (ToS) field in the IPV4 header. The DiffServ field is split into the 6-bit DSCP field and a 2-bit field which is used for Explicit Congestion Notification (ECN) mechanisms as shown in Figure 41.



Bits	7	6	5	4	3	2	1	0
DiffServ Field	DSCP	DSCP	DSCP	DSCP	DSCP	DSCP	ECN	ECN
Function	Class Selector			Drop Probability		0	ECN	

Figure 41: The functions of Differentiated Services (DS) bits

We propose assigning different priorities to different types which affect the treatment of traffic at nodes. Different classes are affected differently by the admission control and the adaptation modules. Highest priority traffic is going to be assured more chance of having resources than lower priority traffic. During congestion, lower priority flows are going to be paused so that high priority traffic flows can have enough bandwidth.

Table 15: Cisco QoS Baseline Classification, Marking, and Mapping

Application	Decimal DSCP	Class of Service	Minimum bandwidth required
IP Routing	48	6	~ 1 KB
Voice	46	5	17 to 106 kbps
Interactive Video	34	4	32-384 kbps
Streaming Video	32	4	20-384 kbps
Locally Defined Mission Critical Data	26	3	< 10 KB
Call Signalling	24	3	~ 1 KB
Transactional Data	18	2	< 10 KB
Network Management	16	2	2kbps
Bulk Data	10	1	10 KB-10 MB
Scavenger	8	1	1 MB
Best Effort	0	0	< 10 KB

RFC 2598 and later RFC 3246 propose another class, Expedited Forwarding (EF) to DSCP classes. The EF class is intended to provide a building block for low delay, low jitter and low loss services by ensuring that the EF aggregate is served at a certain configured rate. We need to identify different types of possible traffic that will be transmitted in MANETs. This traffic needs to be given classes related to the DS Code Points as defined in Table 15. In the technical paper (Cisco, 2005) Cisco defines 11 DSCP based classes in which traffic can be grouped into their systems. The QoS Baseline is a strategic document designed to unify QoS within Cisco. Table 15
Table 15 lists all the classes together with associated DSCP classes allocated to each class.

4.3.2 Admission Control

A new call for transmission needs to be admitted at each node in a path from source to destination of the call. This can only happen if the requested bandwidth (MinBw) of the new flow is smaller than the available bandwidth. The requested bandwidth is as shown in Table 15. Figure 42 demonstrates what happens in the Admission Control Module. When a call for transmission is made by a new flow its requested bandwidth (MinBw) is compared with the (AVbw). The minimum required bandwidth for specific types of traffic is as shown is as shown in Table 15.

If the requested bandwidth is smaller than the available bandwidth then the flow is automatically admitted, otherwise the Bandwidth Adaptation Module is evoked to try to release extra bandwidth being used by other flows so that enough resources are freed to allow the new flow. The Adaptation module returns a modified (AV_{bw}) available bandwidth value which is a sum of the old (AV_{bw}) value together with the released bandwidth (RL_{bw}) values. If the new value of available bandwidth is greater than the minimum requirement of the new flow then it is admitted else the flow is denied.

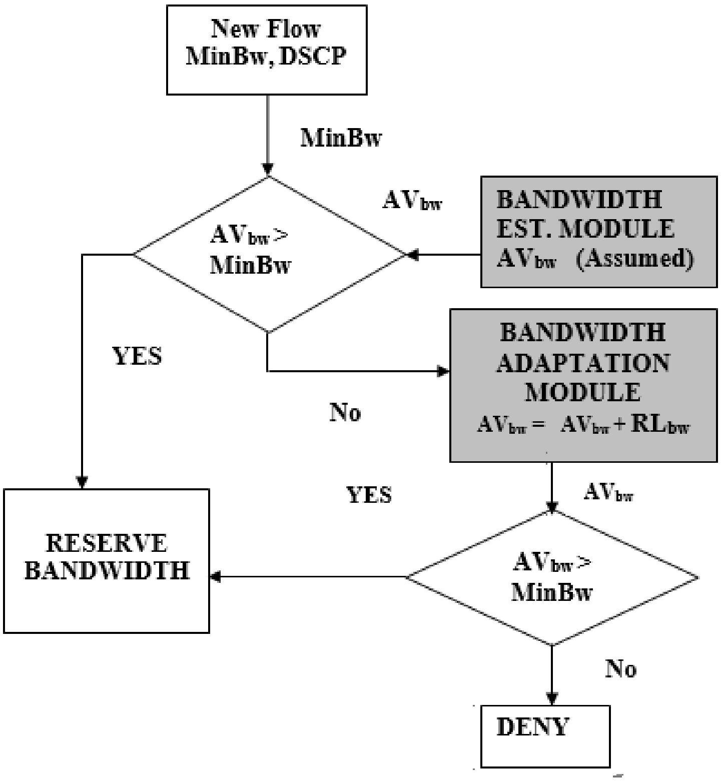


Figure 42: The function of the Admission Control Module

4.3.3 QoS Signalling Method

Our signalling system (AMAN) is an in-band signalling system which uses the packet's header to carry all its control information. It is an adoption of methods in INSIGNIA (S.-B. Lee et al., 2000) and ASAP (Studi et al., 2004). In case of IPv6 this information can be transmitted within the base header and/or within any extension headers. AMAN uses the eight bits of the CLASS field to transmit its Message Type indicator and congestion notification as shown in Figure 43. The IPv6 Hop-by-Hop options extension header will carry the request for reservation (RES), the minimum and maximum required bandwidth (MinBw and MaxBw) and the bandwidth reserved by a node for the specific flow (ActualBw). The structure of the hop-by-hop options extension header is shown in Table 16.

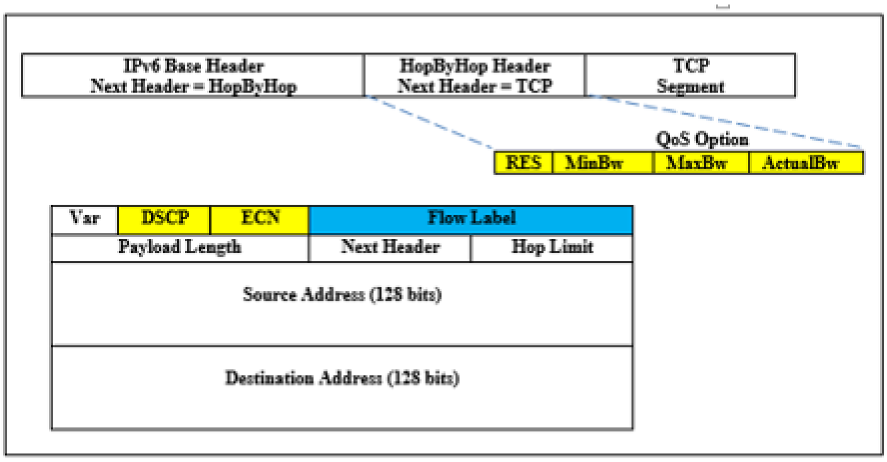


Figure 43: Signalling messages embedded in the IPv6 Header

Table 16: Hop-by-Hop Options extension header format

Octet	0	1	2	3
0	Next Header	Hdr Ext Len	Options and Padding	
32	Options and Padding			
64	Optional: more Options and Padding ...			
96				

We also propose that in case of IPv4 the QoS Signalling information may be carried in the Options field. The QoS option has four fields, the reservation indicator (RES), Minimum and Maximum bandwidth fields (MinBw and MaxBw) and the Actual allocated bandwidth field (ActualBw). The MinBw and MaxBw fields show the minimum and maximum requirements of the traffic. The ActualBw field shows the bandwidth allocated to the traffic by a node. The RES bit is set to 1 if the source is seeking a reservation for a new flow. At every intermediate node, there is an admission control procedure to ascertain whether the resources available can sustain a new flow without hindering on-going communications. When the available bandwidth is not enough, then an intermediate node should not admit new flows, otherwise it will interfere with on-going flows while it does not meet its own minimum requirements.

When the source node initiates a new flow, it fills all the fields with data and forward the packet to the next node. Every intermediate node checks the MinBw and compares it with its available bandwidth and makes a decision whether to admit the flow or not. If there are enough resources (that is AvailBw > MinBw), the node checks if its available bandwidth is greater than the ActualBw in the packet. If the ActualBw from previous node is greater than AvailBw, the ActualBw field is updated with bandwidth equal to AvailBw. An intermediate node can only change the value of ActualBw if its available bandwidth is less than ActualBw. This means that it is a bottleneck.

When the packet arrives at the destination node, various values of ActualBw would have been reserved at nodes along the path. This means that, for some time, there would be over- reservation of resources at some nodes. To correct this, the destination node unicasts a RES report (QR) packet to the source node showing the bottleneck bandwidth for the flow. Figure 44 (S.-B. Lee et al., 2000) shows a source S sending data to destination D along a certain path selected by the routing protocol. All other links have the same value of 200 MB which is reserved at the respective nodes in these links. However node X has 150MB making it a bottleneck node. At node X the ActualBw field will be updated with the value of 150 MB and nodes X and D will reserve 150 MB for the flow.

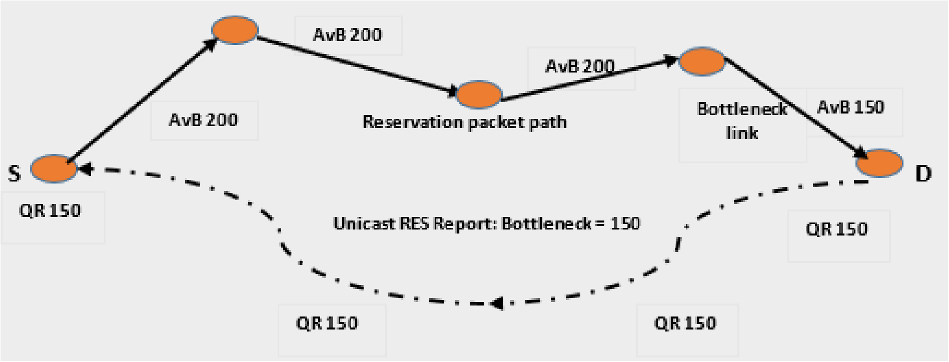


Figure 44: Fast reservation showing request and reservation report

Node D sends a RES report (QR 150) packet to the source node showing the bottleneck bandwidth (150 MB) for the flow. Although it is more efficient, the RES report message does not have to follow the same path and the reservation message since this may not be possible in MANETs. The source will then change the ActualBw field to the value sent by the destination and change the RES field to 0. All the nodes receiving the packets will now allocate this new value ActualBw = 150 MB and release extra bandwidth they had allocated. In summary, reservation packets traverse intermediate nodes executing admission control modules, allocating resources, and establishing virtual path between source/destination pairs. A source node continues to send reservation packets until the destination node completes the reservation setup phase by informing the source node of the status of the flow establishment phase using QOS reporting, as shown in Figure 44.

Every node in the network keeps a QoS table to store information concerning all the flows that are passing through it. The concept of a QoS table is adopted from the authors of ASAP (Xue et al., 2003) as shown in Table 17. The information for each flow includes the *flowID* and the bandwidth allocated to the flow. The *flowID* is identified by the flow label and the source address. The data in the QoS table is updated every time the node receives signalling messages. All this information is embedded in the IP header. This information is used mainly for decision making during reservation and adaptation when the need arises.

Table 17: QoS Table kept at each node keeps details for each flow

Flow Label	Source Address	DSCP	MinBw (kbps)	AvailBw (kbps)	ActualBw (kbps)
1	0122	46	17	200	22
2	1221	34	32	212	45
3	1123	26	28.8	121	40
4	2212	26	28.8	80	38
5	1111	18	12	78	20

4.3.4 Re-routing (Fast restoration).

Reservation-based flows are often rerouted within the lifetime of ongoing sessions due to node mobility, as illustrated in Figure 45. The flow path might be frequently broken, as topology and routing information changes. From the point where the path is broken to the receiver end, no QoS can be guaranteed. To re-establish reservation state on this path as fast as possible is a critical aspect when maintaining the QoS of real-time flows in a MANET.

In such cases, AMAN performs ‘fast restoration’. The goal of restoration is to re-establish reservations as quickly and efficiently as possible. Rerouting active flows involves the MANET routing protocol (to determine new routes), admission control, and resource reservation for nodes along the “new path.” Fast restoration mechanisms also call for the removal of old reservation state at nodes along the “old path.” In an ideal scenario, the restoration of a flow can be accomplished within the duration of a few consecutive packets given that an alternative route is cached.

This type of restoration is called ‘immediate restoration’ (S. ~B. Lee & Campbell, 1998). Each IP packet carries sufficient state information (e.g., service mode and bandwidth request) to establish/re-establish reservations. If no alternative route is cached, the performance of the restoration algorithm is tightly coupled to the speed at which the MANET routing protocols can discover a new path. When a reservation-based flow is rerouted to a new node where resources are unavailable, then the network is in a congested state. The nodes in the subsequent path initiate bandwidth adaptation by releasing bandwidth which was allocated to other flows above their required minimum.

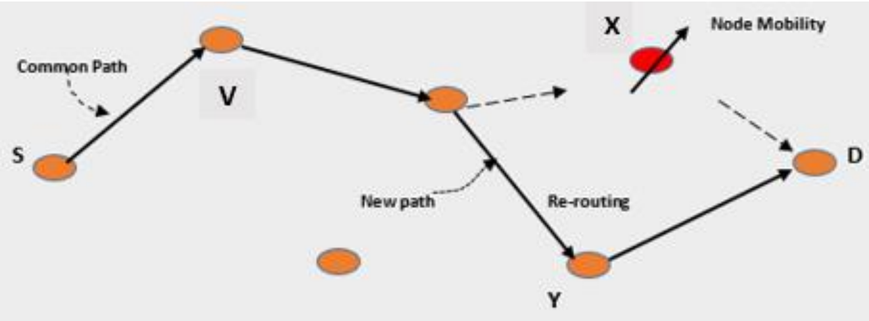


Figure 45: Fast restoration after a link is broken by due to mobility.

Figure 45 (S.-B. Lee et al., 2000) illustrates a fast restoration scenario where intermediate node *X* mov*es* out of radio contact and a reservation-based flow is rerouted through mobile node *X*. The minimum reservation is immediately restored along the new path, while reservations along the old path are timed out and automatically removed. Note that there is no change along the common path*.* We define the common path as any set of hops shared by the old and new paths. Resources that are freed up at nodes along the old path (for example, at *V)* are made available to other flows.

The AMAN system maintains reservations through soft-state resource management. Soft-state timers are continually refreshed and reservations maintained as long as packets associated with a particular flow are periodically received at intermediate routing nodes between source-destination pairs. In contrast, if packets are not received (e.g., due to rerouting or session termination), soft-state timers expire and resources are de-allocated. In the AMAN system, data packets are used to maintain the reservation state at intermediate nodes where the soft-state timer value is automatically coupled to the flow's data rate for optimal performance. A major benefit of our soft-state approach is that resources allocated during the reservation phase are automatically removed in an independent and fully distributed manner when a flow's path changes due to node mobility. For example, resources at *X* in Figure 45 will time out automatically.

4.3.5 Bandwidth Adaptation

This is the novel part of the research. It ensures that a new flow can be incorporated within the network by taking extra bandwidth resources from other applications already in the network. Figure 46 highlights diagrammatically what happens within the Bandwidth Adaptation module. When the request for adaptation arrives the priorities of ongoing applications are checked to get the flow with the least priority.

The adaptation module must degrade the least priority flows to their minimum required bandwidth in order to free resources for the new flow. The module must be able to calculate the amount of bandwidth to be released by low priority flows before degrading them. If the highest priority flow asks for resources when the remaining resources are not enough, then we have to pause some of the least priority traffic to release resources. In case of network congestion notification, AMAN will cause the on-going flows to reduce the allocations (throttle) until the channel is no longer congested. If congestion persists then the least priority flow should be paused. After a random back-off time a source with a throttled or paused flow can attempt to increase or re-admit the traffic flow.

If the minimum required bandwidth, *MinBw*, has been reached for all flows, we have to pause flows to avoid congestion. To avoid reducing reservations for multiple flows in response to mobility-induced congestion, we pause the flows starting with the least priority. We called this state, the panic mode**.** In the panic mode the network should start stopping the flows starting from the least priority flow. Additional flows will be stopped until the system has been restored to a non-congested state.

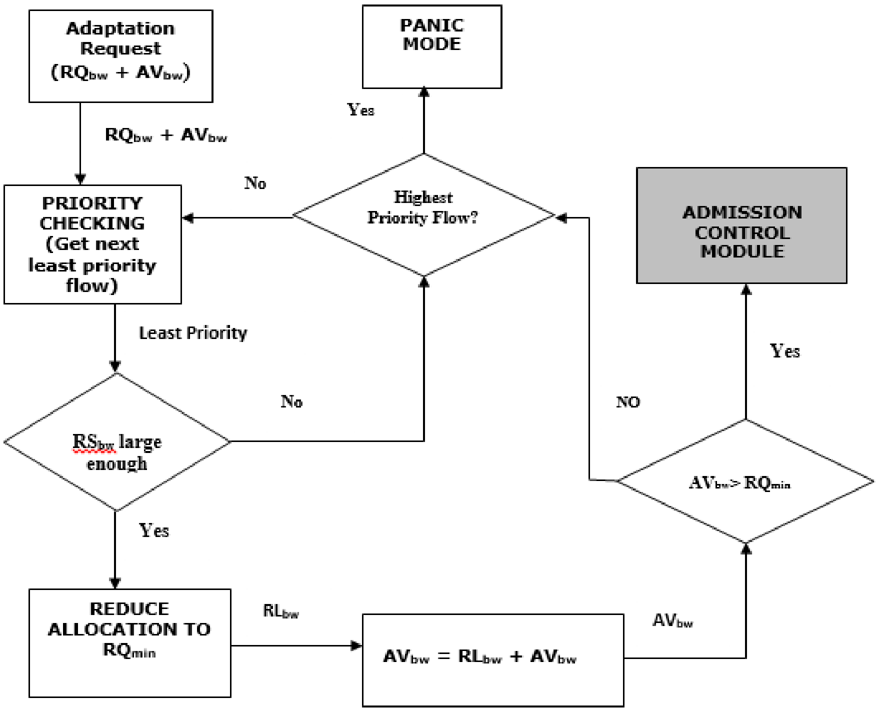


Figure 46: Bandwidth Adaptation module

The QoS framework system supports ongoing end-to-end adaptation that actively monitors network dynamics and adapts flows in response to observed changes based on a laid down adaptation policy. Flow reception quality is monitored at the destination node, and actions are taken to adapt flows under certain observed conditions. The system will always scale up adaptive flows whenever resources become available. The scaling up adaptation process is illustrated in Figure 47. Node mobility or session dynamics may cause a flow routed via Y to be scaled up from minimum to maximum required service. The destination node (D) notes that the *ActualBw* field changes from a lower value to larger value. This indicates that the current path could support higher levels of service.

The destination informs the source of the resource availability via a QoS report. Based on the application's adaptation policy, the source starts to transmit packets with the service mode bit set to 1 and *ActualBw* bandwidth adjusted to the new higher value. This example shows end-to-end adaptation taking place without any change in the current path between the source-destination pair. In this case, end-to-end adaptation is triggered by session-level dynamics (i.e., sessions starting, changing their bandwidth needs, or terminating) rather than mobility conditions.

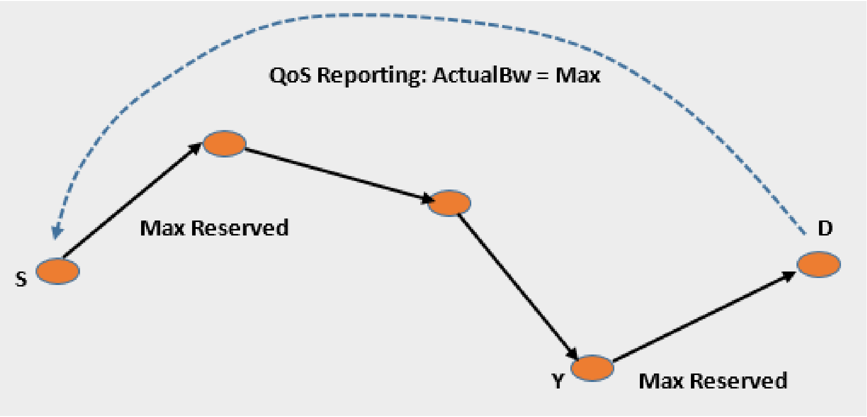


Figure 47: Adaptation: scaling up in case of more resources availed

The final scenario illustrates the scaling down process. In Figure 48 a flow receiving maximum service is rerouted due to the mobility of node *X*. The new path through node *Y* has insufficient resources to support the maximum reserved service. After restoration, the packets are delivered with below minimum bandwidth assurances. The destination node (D) informs the source of this persistent degradation via a QoS report. Following this, the source node (S) scales down and starts transmitting packets at the minimum bandwidth requirement. At the same time *Y* tries to adapt so that all other traffic, passing through it, reduces their reservations to minimum required bandwidth. It does this by changing the value in *ActualBw* field of all other flows to *MinBw* starting with the least priority. It follows the adaptation algorithm of AMAN.

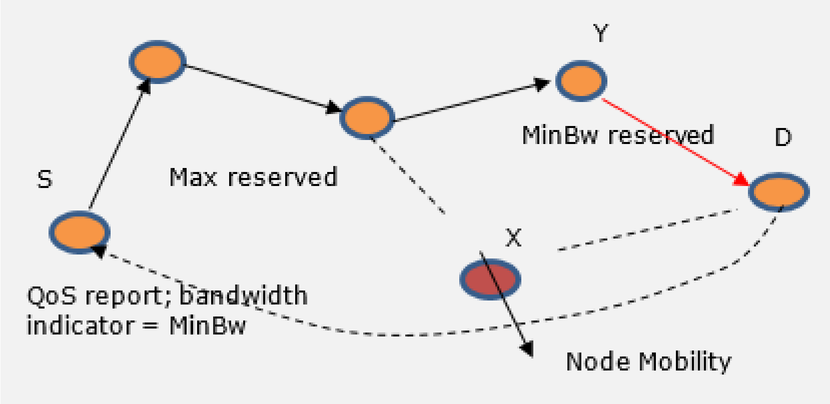


Figure 48: Adaptation: Scaling down in case of resources lost

4.3.6 The Effect of adaptation on the codec

Link Adaptation mechanisms of IEEE 802.11 cause a multi-rate channel problem. In this channel, one user changing its transmission rate provokes a capacity variation of the wireless channel with visible effects for all active calls, like increased delay and packet losses (Sfairopoulou, Bellalta, & Maci, 2008). Contemporary Voice-Over-IP (VoIP) systems typically negotiate only one codec for the entire VoIP session lifetime. However, as different codecs perform differently well under certain network conditions like delay, jitter or packet loss, this can lead to a reduction of quality if those conditions change during the call. In their paper (Aktas, Schmidt, Weingärtner, Schnelke, & Wehrle, 2012) implemented an adaptive strategy that switches the session’s speech codec upon changing network conditions. They designed an adaptive coded switching scheme that depend on available bandwidth. Their adaptive codec switching scheme performs three tasks: (1) choosing the currently best performing codec before the actual communication starts, (2) changing to a low bandwidth consuming low quality codec when the packet loss increases, and (3) changing to a high bandwidth consuming high quality codec when the bandwidth increases. We propose that such codec switching mechanism should be employed together with our framework to stem the effect of adaptation on the codec. We do not discuss codecs any further in this research. We leave it for future work.

4.4 Experimentation and Analysis

The system comprises of a C++ code on which the simulations are done. The code consists of several functions which are called to perform different tasks including computing the available bandwidth. The inputs for the new traffic values are randomised but for simulations that depict real-life situations which generate different classes of traffic. The system code shall be compared with another C++ code, which does not include any bandwidth adaptation.

The code consists of a data structure with five attributes namely:

- *Flow Priority* –Every flow is given a randomised priority value to resemble different real-time traffic, with the flow with the lowest value having the highest priority. These values are important in the Bandwidth Adaptation module when determining which flow(s) to reduce allocation in order to admit a new flow. Flows with the least priorities are reduced first.
- *Flow ID/Address* - Each flow that enters the network has a Flow ID to uniquely identify all flows in the network and also to keep track of the flows which were reduced to required minimum during bandwidth adaptation.
- *Reserved Bandwidth (RBW)* – When a new flow is introduced it comes with a value of its bandwidth requirements. When the flow is admitted

into the network it is given its bandwidth requirements and this becomes the Reserved Bandwidth (RBW).

- *Required Minimum Bandwidth (RQmin)* – This is the minimum required bandwidth requirements of a flow in the network.
- *Releasable Bandwidth (RLBW)* – This is the difference between the reserved bandwidth and the required minimum. The bandwidth that can be released for other new flows to be admitted.

The traffic was defined in terms of class of service of applications. The types of applications, their class of service and required bandwidth were as shown in Table 18. At the beginning of the simulation all the flows are allocated their maximum bandwidth (Max kb/s) and after adaptation they are allocated their required minimum bandwidth.

Table 18: Traffic classes used in the simulations.

Application	Class	Required minimum kb/s	Max kb/s
Voice	5	17	106
Interactive Video	4	32	384
Locally Defined Mission Critical Data	3	5	10
Transactional Data	2	5	10
Bulk Data	1	10	20
Best Effort	0	5	10

A random number of nodes of up to 100 flows is defined at the beginning of each simulations and at set periods of times new flows request admission and this is used for different flows in the simulations. All these values are randomised for the sake of the experimental studies. This random behaviour tries to mimic the random nature of traffic arriving and leaving the node. The Channel Capacity is assumed to be 2Mb. The available bandwidth at any given time will be the difference between the Channel Capacity and the Consumed Bandwidth (i.e. the bandwidth consumed by all the flows in the network). As explained in the model formulation, the new flow goes through admission control and either reservation if the bandwidth requirements are met, or might have to go through bandwidth adaptation first if the available bandwidth is not sufficient. As the simulations are run the results are written onto two files namely *flows* and *flows2*. Both files contain the following attributes:

- Column 1 - FLOW ID
- Column 2 - RESERVED BANDWIDTH
- Column 3 - REQUIRED MINIMUM BANDWIDTH (RQmin)
- Column 4 - PRIORITY

The file *flows2* also contains an additional column of RELEASABLE BANDWIDTH for each flow after Adaptation, which is the difference between the Reserved Bandwidth and the required minimum. This enables us to track which flows were reduced to their working minimum requirements during adaptation. When the code is run (running a simulation) all the flows within that particular simulation are recorded with the above attributes. The file *flows2* can only be written when there is bandwidth adaptation that has taken place. It displays the flows that have been reduced to required minimum. It also displays an entry of the admitted flow if it was successful. If after adaptation the bandwidth requirements of the new flow are not met then there is nothing displayed in the *flows2* file but a denial message is displayed in the flows file.

4.5 Results and findings

Table 19: Load already in the network at the beginning of a simulation.

id	RQmin	RBW	RLBW	priority
1	17	106	89	5
2	32	384	352	4
3	32	384	352	4
4	17	106	89	5
5	32	384	352	4
40	17	106	89	5
41	5	10	5	3

The results in Table 19 show a sample of flows that are in the network at a particular time. The information includes the node id, the required minimum bandwidth (RQmin), reserved bandwidth (RBW) releasable bandwidth (RLBW) and the priority or class of the traffic in the flow. In this sample, the network had 41 nodes of various classes.

Table 20 shows the results of the simulations of flows from Table 9, showing how many flows were admitted or not. These statistics from such tables helped us to extract information, which enabled us to interpret the success rate of the adaptation process, the effect of network load on the admission of a new flow, assessing the impact of reducing the bandwidth allocation of other flows on the admission of a new flow and other data that can be drawn from the results. This network initially had 41 nodes which consumed a total of 4038 KB of data, which is way above the channel capacity. So the network goes through adaptation and 8 flows are reduced and the total consumed bandwidth is reduced to 539KB. The table shows the number of reduced flows, admitted flows and denied flows as time went on up until 1000 seconds elapsed. Hundreds of simulations like this were run to see the general behaviour of the algorithm.

Table 20: Simulation results showing adaptation

Time/s	No of Flows	Reduced	CBW	RBW:	RQmin	Priority:	Admitted	Denied
0	41	0	4038	0	0	0	0	0
5	41	8	4038	0	5	2	0	1
10	42	8	539	10	5	3	1	1
985	179	12	1997	0	17	5	138	68
990	179	12	1997	0	5	2	138	69
995	179	12	1997	0	5	0	138	70
1000	179	12	1997	0	5	0	138	71

There are three things that can happen when a call for admission is made for a new flow. Either a flow is admitted without need for adaptation, adaptation occurs and a flow is admitted or adaptation occurs but still there is no admission of the new flow.

4.5.1 No Adaptation and admission of flow

This happens when the bandwidth requirements of the new flow are met by the available bandwidth and there is no need for Adaptation, thus the flow is automatically admitted into the network. Table 21 shows a sample of the output of a particular simulation, which has 42 flows initially and the consumed bandwidth, CBW = 1.476191MB. With the channel capacity of 2.0 MB the available bandwidth was 0.523809MB. The releasable bandwidth, RLBW, from other flows already in the network is 0.460565MB.

Table 21: Sample results for admission without need for adaptation

Number of flows	: 42
Available Bandwidth	: 0.523809
CBW	: 1.476191
RLBW	:0.460565
New data packet	
Required BW	: 0.152905
Priority	: 1
Minimum	: 0.058824
New packet admitted:	Check network log file
Number of flows	: 43
CBW	: 1.629097
RLBW	: 0.554647

The output also shows the new packet data, which includes the required bandwidth of 0.152905, priority of 1 and a minimum requirement of 0.058824. The flow was admitted automatically without any need for adaptation since the available bandwidth was sufficient to cater for the bandwidth requirements of the new flow.

4.5.2 Adaptation and admission of flow

When the bandwidth requirements of the new flow exceeds the available bandwidth, there is need for reducing allocations of admitted flows. If the minimum requirements of the new flow are met then the flow is admitted. Table

22 shows some highlights of the adaptation process where other flows will have their allocations reduced to their minimum requirements based upon the priorities and then the new flow is admitted once sufficient bandwidth is available.

Table 22: Sample results after adaptation and then admission

Number of flows	: 3
Available Bandwidth	: 0.333333
CBW	: 1.666667
RLBW	: 1.197917
New data packet	
Required BW	: 1.223242
Priority	: 1
Minimum	: 0.784314
Requesting from packet 003 with priority 1 left with 0.005794	

Table 23: Adaptation and no Admission

Number of flows	: 24
Available Bandwidth	: 0.791667
CBW	: 1.208333
RLBW	: 0.407552
New data packet	
Required BW	: 1.376147
Priority	: 1
Minimum	: 0.431373
Flow denied: Bandwidth not sufficient	

4.5.3 Adaptation and no admission of flow

When the bandwidth requirements of the new flow exceeds the available bandwidth and adaptation is done but the bandwidth requirements cannot be met even after all the flows have been reduced to their working minimum then the flow is denied. A sample output of a denied flow is shown in Table 23.

4.5.4 Analysis of results

Data from the simulations help us to calculate the success rate of bandwidth adaptation in the resource reservation scheme. The success rate of the Bandwidth Adaptation process gives us a rough probability that when a new flow is initiated in a loaded network, the flow is admitted. We compared our results from adaptation simulations with a control experiment without adaptation for the purposes of comparison. We estimate the admission success rate of a scheme as:

$$Success\ Rate\ of\ Admitance = \frac{No.\ of\ flows\ admitted}{Total\ No.of\ flows}\ .100\ \% \qquad 4.1$$

The simulation results highlights that the Adaptation process increases the chances of admission of flows that could have been denied by a factor of 0.73 or success rate of 73% as shown by the estimates of the Success rate of a scheme with bandwidth adaptation.

The simulation results of a system without bandwidth adaptation, the number of flows that were admitted reveal that without adaptation the admission rate is far much lower at 40%. The Adaptation results show that the efficiency of the Adaptation process, as represented by the number of flows admitted after Adaptation process is increased by a factor of 0.6 or success rate of 60%. Success rate of adaptation process which is determined by the number of flows admitted after adaptation has taken place was found to be 73 %. Success rate of a new flow to be admitted without adaptation was found to be 40% as shown in Figure 49. This shows primarily that adaptation improves the chance of admitting a new flow by more than 30%. We also look at the relationship and the effect of number of flows in the network on the admission of a new flow and on the number of flows that can be reduced to working minimum.

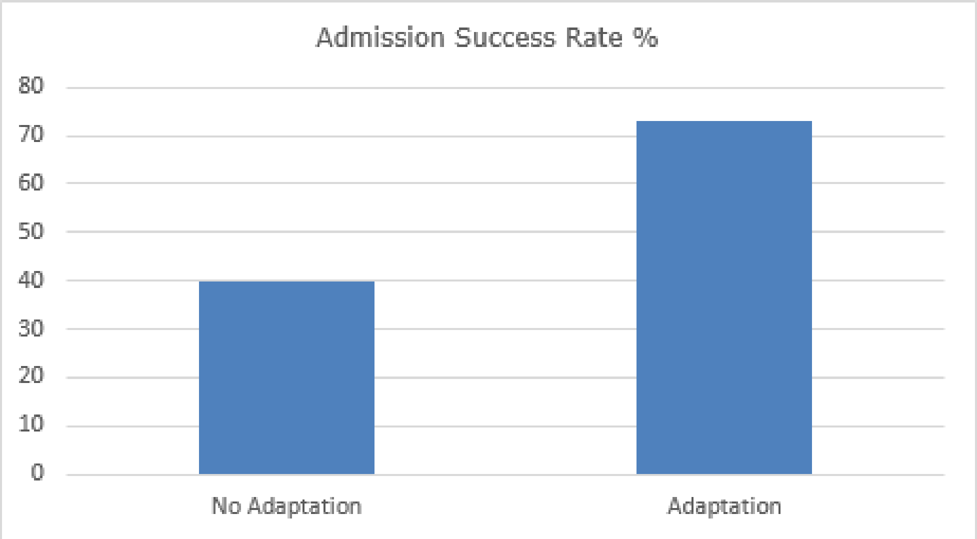


Figure 49: Admission success rate of adaptation Vs without adaptation.

Table 24 and Figure 50 show the variation of number of nodes whose bandwidth allocations reduced to their minimum required bandwidth with network size. The results show that the number of reduced nodes is exponentially proportional to the number of flows that were present in the network at the beginning of the simulation.

Table 24: Variation of nodes adapted downwards with iniitial network load

nodes in network	Average number of nodes adapted downwards	standard deviation
0	12.1	3.1
4	14.4	4.3
12	18.2	5.2
18	22.3	3.4
28	31.5	4.4
33	36	5.1
41	44.1	4.1
72	77.3	3.1
91	112	3.5
99	134	4.3

The general trend highlights that as the network size increases the number of flows that will be reduced to their minimum also increases exponentially. This is an expected result, since when the number of nodes is low, then there is a lot of available bandwidth hence there is no need for adaptation. However as the number of nodes in the network is large then the available bandwidth is low and more and more flows have to be adapted to release bandwidth for new flows.

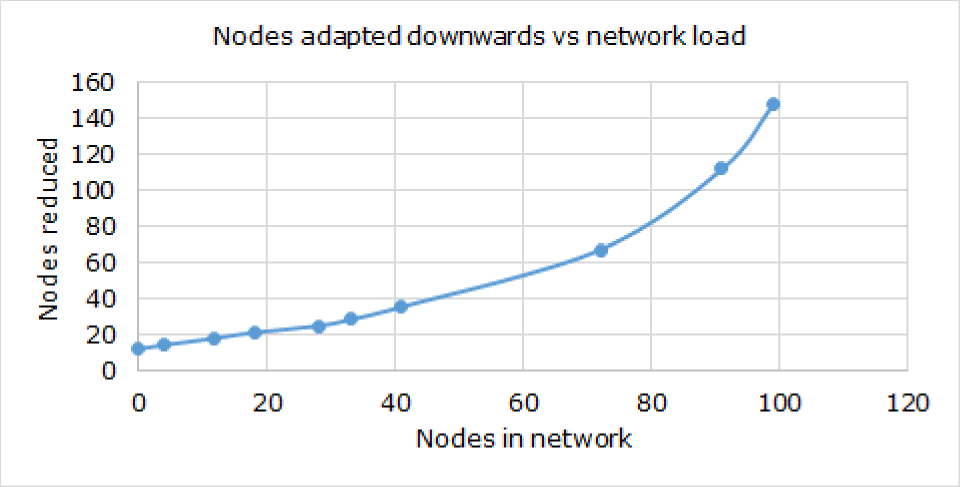


Figure 50: The variation of number of flows reduced with load increase

Although adaptation might have a negative effect on the quality of service of the flows reduced performance will remain within acceptable range since the adaptation does not go below the minimum required bandwidth. Under congestion, in case of rerouted flows, adaptation and flow pausing might impact heavily on the network performance since some applications will be forced to stop operation. The standard deviation shows that the dispersion of values is generally small indicating the reliability of the values used in the calculation.

Results in Table 25 and Figure 51 show the average number of nodes admitted in relation to the number of nodes that were present in the network at the beginning of the simulation. The relationship is generally inversely proportional to the number of nodes in the network. A huge number of nodes was admitted for low network load and the number decreased when the network load increased.

Table 25: Variation of admitted nodes with initial network load

Number of nodes in network	Average number of nodes admitted	Standard deviation
0	168.08	6.84
4	162.18	10.33
12	156.41	10.79
18	146.75	6.22
28	138	13.28
33	135.41	9.26
41	130.5	6.23
72	96	11.19
91	71.9	11.91
99	65.75	9.73

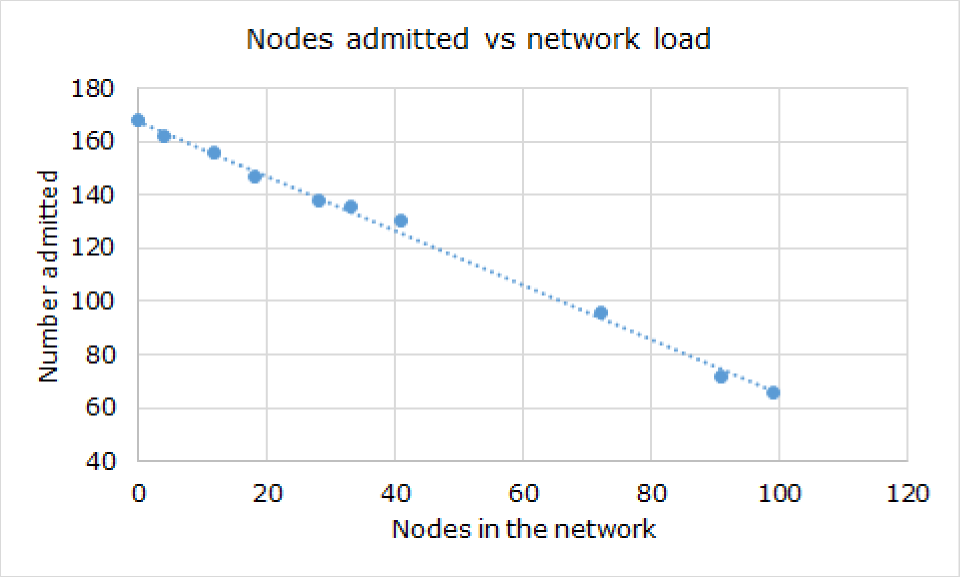


Figure 51: Variation of admitted nodes with initial network load

Results in Table 26 and Figure 52 show variation of the number of nodes that whose flows were denied access to the network as we varied the number of nodes in the network.

Table 26: Variation of number of flows denied access with network load

Number of nodes in network	Average number of nodes denied	Standard deviation
0	30.08	10.09
4	37	12.32
12	48.83	12.58
18	52.71	9.11
28	55.7	20.1
33	60.75	11.46
41	64.83	11.69
72	96.58	22.19
91	110.75	33.7
99	129.17	9.879

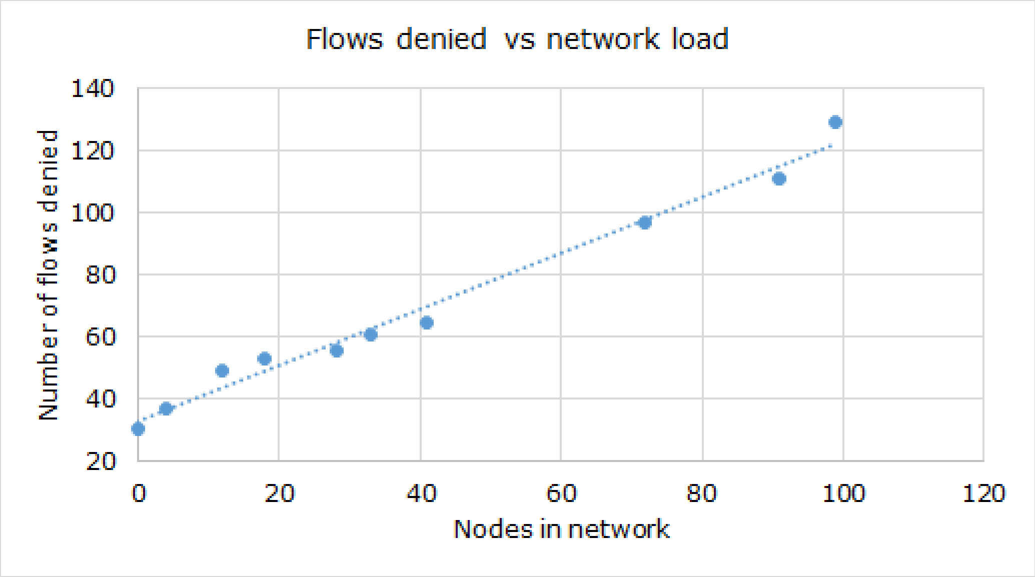


Figure 52: Variation of nodes denied access with the network load

In general, the number of nodes denied increased almost linearly with the network load. We can attribute this characteristic to the fact that available bandwidth becomes less and less as the size of the network increases. So the number of number of nodes denied at the end of the simulation if bigger for a big network than for a smaller network. The values of standard deviation are generally big. We can attribute this to the fact that the classes of flows in the network varied greatly from one simulation to the next since they were randomly created.

Figure 53 shows the relationship between the number of flows in a network and the consumed bandwidth. For the network that does not do adaptation the number of nodes remains very small as the bandwidth consumed increases. This is attributed to the fact that flows will continue consuming bandwidth close their maximum requirements taking no regard to the congestion in the network.

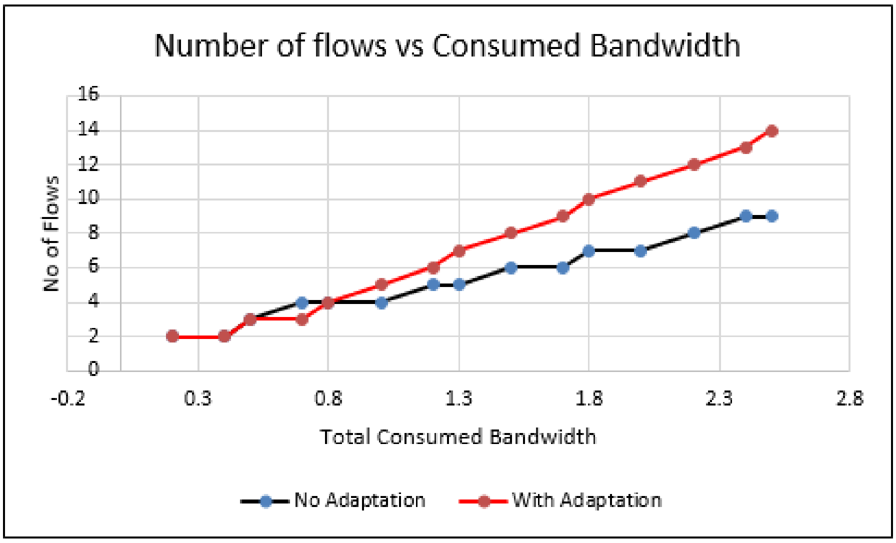


Figure 53: Number of flows against the consumed bandwidth.

The network with adaptation allows more flows to be admitted for whilst consuming less bandwidth.

4.6 Conclusion

In this chapter we presented a scheme for traffic differentiation and management in Mobile Ad Hoc Networks (MANETs). The proposed scheme is intended to efficiently manage the reservation of bandwidth in MANETs based upon the available bandwidth within the network. The scheme also employs the bandwidth adaptation process, which is the novel part of the research, to increase the chances of admission of a new flow into the network and to control congestion. We have discussed in detail and proved by simulation how the proposed scheme works. The scheme increases the available bandwidth in the network by reducing the reserved bandwidth of other flows to their minimum requirements according to the priorities of the flows thereby allowing more flows to be admitted. The research also includes an empirical analysis of the behaviour of the scheme by conducting simulations under varying conditions of required bandwidth and priority of new flow, available bandwidth and the size of the network. The results presented for the proposed scheme demonstrates that Bandwidth Adaptation enhances the admission of flows with bandwidth requirements greater than the available bandwidth.

It is difficult, at this point, to do a qualitative comparison of this scheme with other schemes like ASAP, SWAN and INSIGNIA because this scheme is not adequate since it needs to be embedded in an ad hoc routing protocol, then re-implement and simulate in NS-2 Simulator. As a result, this scheme will only be compared against a scheme without Adaptation in order to assess the efficiency of the Bandwidth Adaptation process. However our scheme has an added attribute of multi-level priority scheme which gives a pre-emptive advantage to the highest priority traffic. In other schemes like INSIGNIA, ASAP and SWAN, traffic is differentiated into real-time and best-effort traffic only. This does not respect the idea that multimedia traffic that is flourishing in the internet these days have varying characteristics and varying importance. There is also a need to test the framework in a real life test bed situation, so that the results obtained here from simulations and the benefits therefore expected, can be verified and quantified. The scheme has the potential of coming up with efficient and realistic reservations, which are comparable to those of other frameworks, like ASAP and SWAN.

5. Time-Slot Assignment in TDMA MANETs[4]

5.1 Problem definition

The MAC sublayer converts raw physical capacity into usable network capacity, and thus the choice of a MAC protocol significantly impacts MANET performance. Approaches to Medium Access Control (MAC) in mobile ad hoc networks (MANETs) can be broadly classified into Time Division Multiple Access (TDMA) and Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). In today’s MANETs CSMA/CA variants are generally preferred, in particular, the IEEE 802.11 DCF. This is partly due to the cheap availability of IEEE 802.11 cards, and partly due to the fact that its simplicity, robustness and flexibility are a ready fit for MANETs. Also because of a need to provide network-wide synchronization without centralized control, and to accommodate mobility makes TDMA very hard to design and implement. It is believed that TDMA offers superior performance as well as better capacity guarantees compared to CSMA/CA. While the proliferation of real-time multimedia applications demands a protocol with TDMA-like features, the problem of doing so in a practically viable manner remains unsolved (Jakllari & Ramanathan, 2009)

Time Division Multiple Access (TDMA) is widely acknowledged as being an excellent fit for providing requisite QoS for realtime applications, as it enables allocation of dedicated channel capacity to flows. However, broadly speaking, TDMA needs two things that IEEE 802.11 does not, which are, synchronization of frames and slots, and allocation of slots to nodes/links. In MANETs, unlike in cellular networks, the lack of centralized control and mobility of nodes makes

[4] Work presented in this chapter is an extension of work presented in the following conference proceedings; **B. M Nyambo**, G.K. Janssens and W. Lamotte (2012). Quality of service in TDMA MANETs using prioritised time slot assignment. In: Vladimir Janousek and Sarka Kvetonova (eds.), *Proceedings of the Industrial Simulation Conference’2012, Brno, Czech Republic*, 4-6 June 2012, pp. 217-222 (ISBN 978-90-77381-71-7).

both of these extremely hard. Even when solved, the need for guard times between slots and control messages for allocation both lead to low efficiency that negates the advantages provided by TDMA over IEEE 802.11. Furthermore, synchronization is inherently not scalable with network size. Despite numerous efforts, the problem of providing practically viable solutions to these two challenges has not been satisfactorily solved, resulting in the community being stuck in sub-optimal solutions.

Our work described in Chapters 3 and 4 is based on the CSMA/CA based IEEE802.11 MAC protocols like DCF. We also need to expand the principle of differentiated traffic with priority based classes to the slot allocation algorithms in TDMA. In this chapter we design a distributed reservation system in TDMA based MANETs. We propose a prioritised time slot assignment algorithm that runs on top of a routing algorithm to make a solid TDMA reservation system.

5.2 System definition

Time division Multiple Access (TDMA) is a channel access method that allows several users to share the same frequency by dividing time into time frames and each user is allocated a time slot within each time frame. Figure 54, illustrates the concept of splitting time into time frames and time slots in TDMA (T. Lee & Park, 2001). It finds use in telecommunications fields like satellites, telephones and wireless networks (Jawhar & Wu, 2008).

In this model, a node can broadcast a message to its adjacent nodes. The time is divided into a non-overlapping equal time period time frame which is divided into a number of non-overlapping equal time periods, called time slots. The slots are numbered from 1 to *MaxSlot*. We assume that *MaxSlot* is sufficiently large to handle all the assignment strategies for an input graph. Informally, the objective of the scheduling is that each node picks a time slot during which it can transmit without conflict. We say that two nodes are in conflict if and only if they are both using the time slot and are one or two hops away from each other. Typically this definition of conflict is used in TDMA scheduling where two nodes within a two hop range have interference at some node due to the hidden terminal problem.

This interference causes that node to receive a degraded in received signal (Rhee, Warrier, Min, & Xu, 2009).

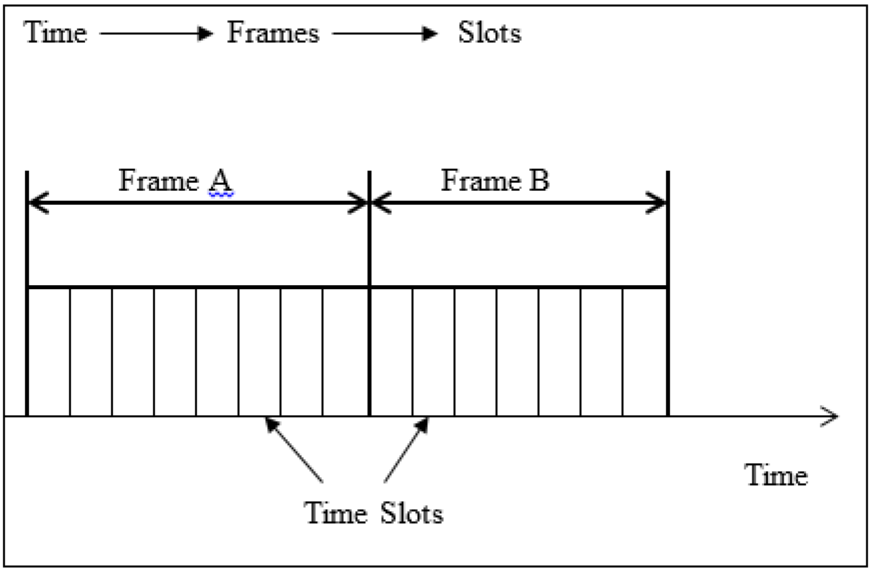


Figure 54: Division of time into frames and time slots.

We need to find a scheduling algorithm that will allow us to schedule traffic by optimally allocating time slots in TDMA for a MANET all this in close association with a routing protocol to find a QoS path. The number of unused slots is supposed to be minimized. The algorithm should take into account the different priorities in the traffic, giving higher priority traffic more slots. There is a need that if the highest priority traffic requests bandwidth and the channel is busy and all time slots are in use, the algorithm should release time slots allocated to the least priority flow and allocate them to the new traffic.

5.2.1 Formulation of problem as a graph

A MANET can be modelled by a directed graph *G = (V, E)*, which is a bipartite multigraph, where *V* is the node set and *E* is the edge set of *G*. The nodes denote stations in a network. *E* is such that for any two distinct nodes *u* and *v*, edge (*u*, *v*) $\in E$ if *v* can receive transmission from *u*. We assume that (*u*, *v*) $\in E$ if and only if (*v, u*) $\in E$, that is links are bidirectional. From a graph theoretical point of view, scheduling in TDMA-based MANETs is equivalent to distance-2 colouring of a *G= (V, E).* Distance-2 neighbours of a node include all its 1-hop and 2-hop neighbours. The corresponding problem is to produce an assignment of colours such that no two are assigned the same colour if they are distance-2 neighbours (Jawhar & Wu, 2008).

We define the TDMA scheduling problem to be a problem of allocating time slots for each node, given an input graph, such that if any two nodes are in conflict, they do not have the same time slot. We say that two nodes *u* and *v* are in conflict if and only if *u* and *v* are in one or two hops away from each other.

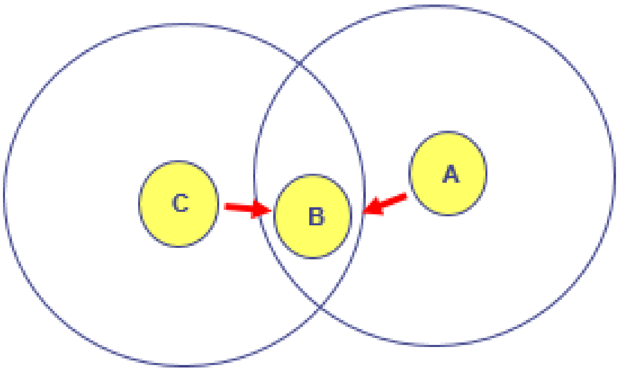


Figure 55: Illustration of nodes in conflict

Typically this definition of conflict is used in a broadcast mode of TDMA scheduling where any two nodes within a two hop range can have radio interference at some node in their transmission ranges due to hidden terminal problems and their radio broadcast transmission causes that node to receive graded signals as shown in Figure 55.

This TDMA scheduling problem is often known as the static channel assignment problem or reuse channel assignment problem. After each node finds its slot, it (re)uses that slot at each time frame for collision-free data transmission. Thus, an algorithm that minimizes the number of time slots being assigned allows the system to minimize the frame size (originally set to *MaxSlot*), thus increasing channel utilization. After the channel assignment, the maximum time slot being assigned in the network must be broadcast to the entire network. Our definition of the TDMA scheduling problem deals only with the channel assignment part (Rhee et al., 2009).

In the TDMA model, a node's use of a slot depends not only on the status of its 1 -hop neighbours' use of the slot but also on the 2-hop neighbours' current use of the slot. This is due to the hidden and exposed terminal problem which must be taken into account. This means that a slot is free if and only if no other node in the 2-hop neighbourhood is using it. In a fully distributed broadcast scheduling algorithm, each node calculates its own schedule based on its information and information from its 1-hop and 2-hop neighbours. Schedules are computed in a parallel fashion, making such algorithms more scalable and practical than centralized scheduling.

The MAC sublayer converts raw physical capacity into usable network capacity, and thus the choice of a MAC protocol significantly impacts MANET performance. Approaches to Medium Access Control (MAC) in mobile ad hoc networks (MANETs) can be broadly classified into TDMA and CSMA/CA. In today's MANETs CSMA/CA variants are generally preferred, in particular, the IEEE 802.11 DCF. This is partly due to the cheap availability of IEEE 802.11 cards, and partly due to the fact that its simplicity, robustness and flexibility are a ready fit for MANETs. Also because of a need to provide network-wide synchronization without centralized control, and to accommodate mobility, TDMA is very hard to design and implement. It is believed that TDMA offers superior performance as well as better capacity guarantees compared to CSMA/CA. While the proliferation of real-time multimedia applications demands a protocol with TDMA-like features, the problem of doing so in a practically viable manner remains unsolved.

Time Division Multiple Access (TDMA) is widely acknowledged as being an excellent fit for providing requisite QoS for real-time applications, as it enables allocation of dedicated channel capacity to flows. However, broadly speaking, TDMA needs two things that IEEE802.11 does not need. These are synchronization of frames and slots, and allocation of slots to nodes/links. In MANETs, unlike in cellular networks, the lack of centralized control and mobility of nodes make both of these extremely hard. Even when solved, the need for guard times between slots and control messages for allocation both lead to low efficiency that negates the advantages provided by TDMA over IEEE802.11. Furthermore, synchronization is inherently not scalable with network size. Despite numerous efforts, the problem of providing practically viable solutions to these two challenges has not been satisfactorily solved, resulting in the community being stuck in sub-optimal solutions. Work described in Chapters 3 and 4 is based on the CSMA/CA based IEEE802.11 MAC protocols like Distributed Coordinated Function (DCF). We also need to expand the principle of differentiated traffic with priority based classes to the slot allocation algorithms in TDMA.

5.2.2 Time slot assignment and QoS routing

In this section, a DSR-based on-demand QoS routing protocol designed by (Jawhar & Wu, 2004) is extended to include prioritised pre-emptive time slot assignment. The implementation of the protocol assumes a TDMA synchronous networking environment. In this network, communication between nodes is done using a synchronous TDMA frame. The TDMA frame is composed of a control phase and a data phase.

5.2.2.1 Basics

Figure 56 (Liao, Tseng, & Shih, 2002), shows the TDMA frame structure for a TDMA network (or a TDMA cluster) of N nodes. Each node in the network has a designated control time slot (control slots 1 through N in this example), which it uses to transmit its control information, but the nodes in the network must compete for use of data phase time slots.

In order to prevent interference in the TDMA environment, a time slot *t* is considered free to be allocated to send data from a node *x* to a node *y* the following conditions are true (Liao et al., 2002):

1) Slot t is not scheduled for receiving or transmitting in neither node *x* nor *y*.
2) Slot t is not scheduled for receiving in a node *z* that is a 1-hop neighbour of *x*.

3) Slot t is not scheduled for sending in any node *z* that is a 1-hop neighbour of *y*.

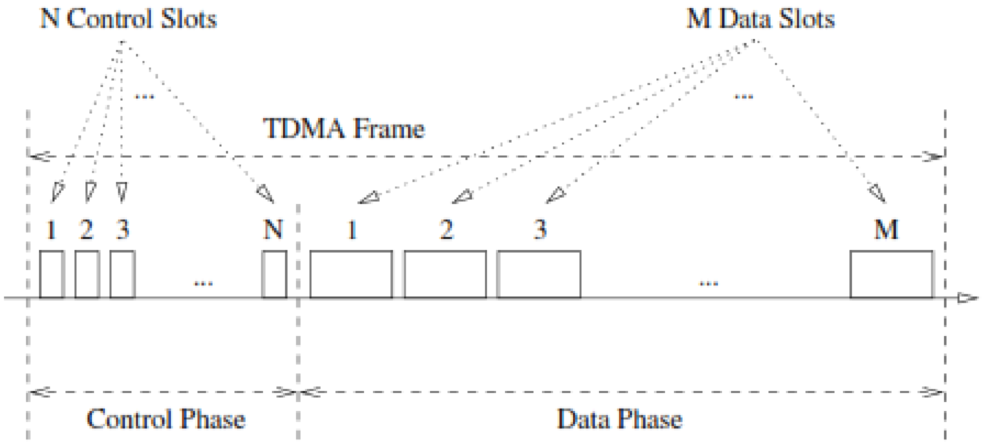


Figure 56: TDMA frame is split into control phase the data phase slots

5.2.2.2 The Dynamic Bandwidth Reservation Protocol

In (Jawhar & Wu, 2005) they present a dynamic range bandwidth reservation protocol for wireless networks. It is on-demand, source based and similar to DSR (Perkins, Royer, Das, & Marina, 2001). Its on-demand nature makes it generally more efficient, since control overhead traffic is only needed when data communication between nodes is desired.

5.2.2.3 The data structures

Each node maintains and updates three tables, slot status table (*ST),* receive table *(RT)* and neighbourhood table (*H)*. At a node *x*, the tables are denoted by ST_{x}, RT_{x} and H_{x}. A slot can be free (0), allocated to send (1) or reserved to receive (2). The H_x table contains information about which nodes are 1-hop and 2-hop neighbours of *x*.

- ST_x *[1...n, 1...s]:* This is the send table which contains slot status information for the 1-hop and 2-hop neighbours. For a neighbour *i* and slot *j*, $ST_x[i, j]$ can have one of the following values representing two different states: 0 - for free, and 1 - for reserved to send.

- *RT_x [1...n, 1...s]*: This is the receive table which contains slot status information for the 1-hop and 2-hop neighbours. For a neighbour i and slot *j*, *RT_x[i, j]* can have one of the following values representing two different states: 0 - for free, 1-for reserved to receive.
- *H_x [1...n, 1...n]*: This table contains information about node *x*'s 1-hop and 2-hop neighbourhood. If an entry *H_x[i, j]* equals 1, this means that node *i*, which is a 1-hop neighbour of node *x*, has node j as a neighbour; an entry of infinity indicates that it does not.

5.2.2.4 The algorithm at the source

Figure 57 (S.-B. Lee et al., 2000) illustrates the signalling algorithm from the source node to the destination node. When a source node *S* wants to send data to a destination node *D* with a bandwidth requirement of minimum, b_{min}, and maximum, b_{max}, number of slots, it initiates the QoS path discovery process.

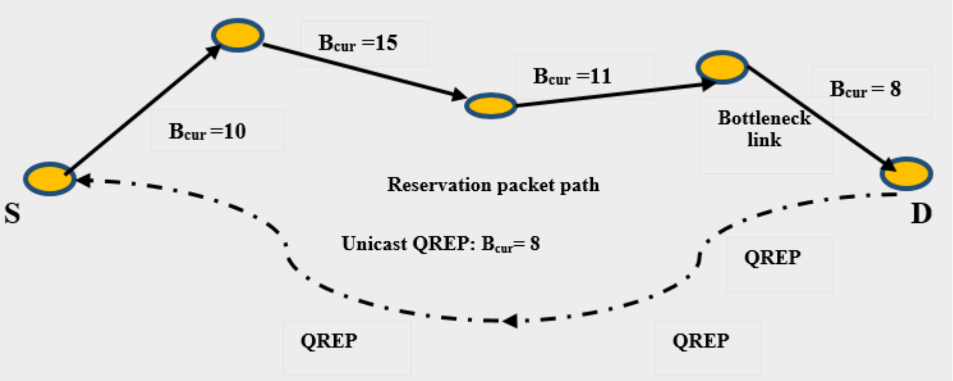


Figure 57: Path set up between a source S and Destination D

S determines if it has enough slots, in the specified range, to send from itself to each one of its 1-hop neighbours before it broadcasts a quality of service request message *QREQ(S, D, id, b_{min}, b_{max}, x, PATH, NH, P)* to all of its neighbours. The message contains the following fields:

- *S* is the source ID*, D* is the destination ID and *id* is the session ID*.* The values *(S; D; id)* are unique for every *QRQ* message and are useful in preventing looping.

- b_{min} and b_{max}: represent the minimum and maximum number of slots required for the session and have to be present in the path from *S* to *D*.
- *x*: The node ID of the host that is forwarding this *QRQ* message.
- *PATH*: is a list of the form *(h_1, l_1), (h_2, l_2) ... (h_k, l_k)*) containing the accumulated list of hosts and time slots, which have been allocated by this *QRQ* message so far. The variable h_i is the i^{th} host in the path and l_i is the list of slots used by h_i to send to h_{i+1}
- *NH*: is the next hop list which is in the form *((h'_1, l'_1, $b'_{1\_ cur}$), (h'2, $l'_{2,}$ $b'2\__{cur}$)... (h'_k, l'_k, b_{k_cur}))*. The values *(h'_i, l'_i, b_{i_cur}*) represent the ID of the host, a list of the slots which can be used to send data from *x* to h'_I , and the current number of allocated slots in the QoS path (so far) from *S* to *D* (passing through *h*) as the path is being allocated.
- *P*: is the priority of the traffic in the flow. Traffic will be grouped into many priorities and high priority traffic will be given more preference whenever resources are limited.

5.2.2.5 The algorithm at an intermediate node

The *QRQ* message will travel from source to destination through intermediate nodes. Any intermediate node *y* receives the *QRQ* message, *y* checks the tuple *S/D/SessionID* to confirm that this message was not processed previously otherwise. If so, this *QRQ* message will be dropped, to prevent looping. If this *QRQ* message was received for the first time, the intermediate node performs the following algorithm:

i. *Retrieve b_{cur} and update the ST and RT tables*

ii. *Determine to which neighbours the QRQ message must be propagated (to include in the NH list)*

iii. *If* y *does not have enough slots with any neighbour then try to downgrade another path*

iv. *If the constructed NH list is not empty then forward the QRQ message*

v. *If the constructed NH list is empty, decide if it is possible to place the QRQ message in the QRQ pending queue or drop it*

5.2.2.6 Sending the reply message from the destination

When the two *QRQ* messages arrive at the destination node *D*, this indicates that the QoS path from *S* to *D* with b_{cur} slots where $b_{min} \leq b_{cur} \leq b_{max}$ in each hop has been discovered. The destination *D* unicasts a *QREP(S, D, id,* b_{min}*,* b_{max}*,* b_{cur} *PATH, NH)* to the source through all of the intermediate nodes that are specified in *PATH.* This confirms that the path was allocated by the corresponding *QRQ* message. *PATH* contains a list of the nodes along the discovered path along with the slots which were allocated for this path at each node. Intermediate nodes, upon receiving the *QREP* will reserve b_{cur} slots and will free any additional slots that were allocated for this path. When the *QREP* message arrives at the source node S, it will then initiate data transmission along the reserved path using b_{cur} data slots per frame where $b_{min} \leq b_{cur} \leq b_{max}$

5.2.2.7 FAST Restoration

Due to the nature of MANETs, some nodes in the selected path may move out of transmission range of its neighbours and the path is broken. In order to come up with a fast restoration process, the nearest node will try to find a new path with enough resources. Figure 58 (S.-B. Lee et al., 2000), illustrates re-routing in case of mobility. When node *M* moves beyond the transmission range of *X*, node *X* has to look for an alternative route to node *D* but at the same time maintaining the common path. In this way the process of route discovery will be faster since the common path from *S* up to *X* is maintained in the routing table.

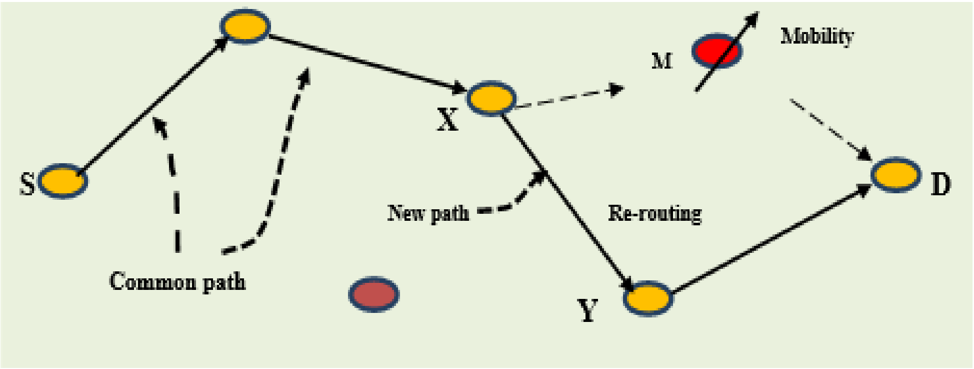


Figure 58: Re-routing have to be initiated after a path break

5.2.2.8 Adaptation Due to congestion

The new path found after fast restoration may not have enough resources as the previous one so the nodes should try to degrade their flows so that they can release some slots and remain with at least b_{min} slots for each flow. If congestion persists, then the network should start halting flows starting with the least priority flow. The nodes will use the Prioritised Dining Philosophers algorithm as described in section 5.5. In this case the least priority flows will be pre-empted of their flows by high priority flows so that b_{min} for high priority flows will be preserved. The research in this chapter is based on the prioritised dining philosophers' algorithm of slot allocation.

5.3 Model formulation

In Rhee et al. (2009) they used a modified dining philosophers' algorithm called DRAND to solve the assignment problem in MANETs. The Dining Philosophers (DP) problem is a classical resource allocation problem that formulates a common synchronization need of multiple processes in accessing a set of exclusive resources. The DP problem can be defined in the following way. There are *n* philosophers in the system and a chop-stick set *F.* Each philosopher rotates its state from *thinking*, *hungry*, *eating* and *releasing*. In order to eat, each philosopher needs a fixed set of chop-sticks (a subset of *F*), and it needs to acquire all of them to start eating. We say that two philosophers are *contending* if their chop-stick sets contain a common Chop-stick. When finished eating, he releases his chop-sticks for use by his contenders or by himself later when he becomes hungry again. No two contending philosophers can eat at the same time. The goal is to minimize the waiting time of hungry philosophers (also known as *response time*). The DP problem captures the type of synchronization and resource allocation requirements commonly arising in distributed systems such as database transaction systems and distributed file systems where multiple processes need to update several data items or files "consistently" at the same time.

In this research we apply the DP to the timeslot assignment problem in TDMA based MANETs. We reformulate the problem in the following way. There are *n* nodes in the system and a frameset *F* which is composed of a set of time slots *s*. Each node rotates its state from *Idle, Trying, Slot usage and Releasing*. In order to transmit, each node requires a fixed time slot *s* which is a subset of the frameset *F*. It needs to acquire the time slot in order to start transmitting. We say two nodes are contending if they require use for one time slot simultaneously. When finished transmitting or receiving, the node releases the time slot for other nodes to use or for use by itself later when it requires channel usage again. No two contending nodes can use the time slot at the same time. The goal is to minimise waiting time for high priority channels

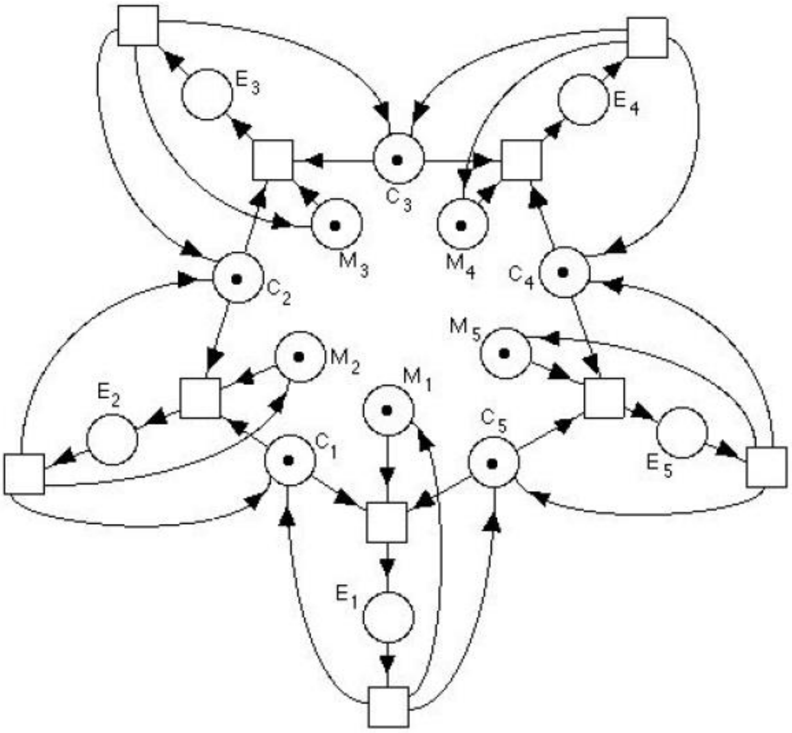


Figure 59: Petri net for dining philosophers' problem. (Mcguigan, 2007)

A frameset *F* is composed of the time slots s which is equivalent to the components of the chop-stick set. It this comparison, Node (*n*) =Philosopher (*p*), frameset (*F*) =chop-stick set (*F*) Slot Usage= Eating (which is either transmitting or receiving), Thinking=Idle time, Hungry=Trying, Releasing

=Releasing. We can model the DP model in MANETs using Petri Nets as shown in Figure 59. The places in the Petri net represent states of a philosopher *i*, for example M_i with a token indicating that the philosopher *i* is meditating and E_i means that he is eating. The remaining places (indicated by C_j) represent resources (in this case time slots)

5.3.1 Prioritised Dining Philosophers Algorithm

In our solution we need to factor in priorities within nodes such that those nodes with multimedia traffic get a higher priority than Best-effort traffic. We call the algorithm prioritized Dining philosophers Assignment (PDPA) algorithm. Initially the node is in an idle state, which is similar to the thinking state in the dining philosopher's algorithm. If the node decides it now wants to receive or transmit, it sends a request and changes from the idle state to the trying state. If we draw parallels to the DP algorithm, this is similar to the Hungry state of the philosopher. If the node succeeds in gaining access to the channel it gets connected and graduates into the slot usage state.

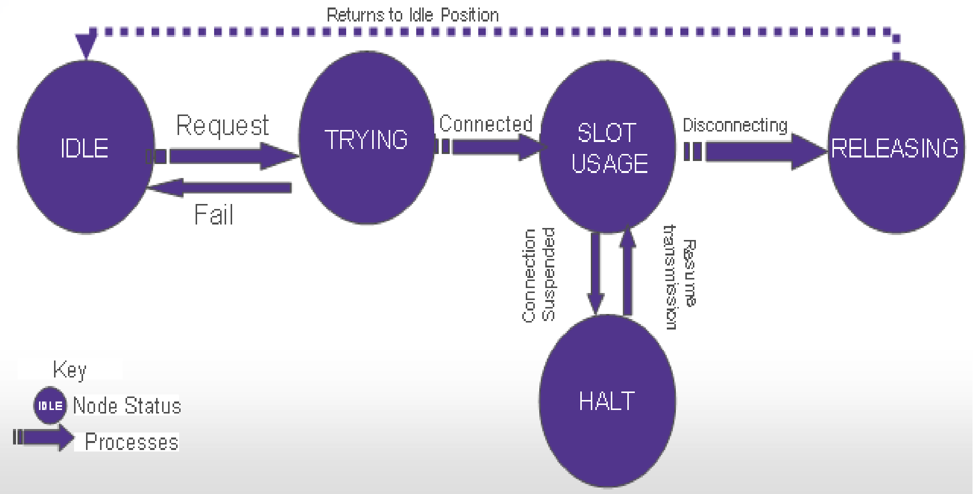


Figure 60: Variation of states in the PDP assignment algorithm

A race condition exists between the contending nodes based on the priorities assigned on each node. The outcome depends upon which of two or more competing processes is granted a resource first. In the worst case scenario, if there are no more free slots and a higher priority node requests slot usage whilst a lower priority node is using it, the algorithm allows the higher priority traffic containing node to take precedence over the lower priority node. The lower priority node will temporarily go into a suspense state (HALT STATE). If the higher priority nodes are done the lower priority will continue with the transmission. To avoid the case of lower priority being locked out in the Halt state, the algorithm promotes the suspended process to the highest priority so that it is not suspended again once it comes back on-line. This avoids starvation of lower priority traffic. Figure 60 shows the variation of the states of a node.

5.3.2 PDP priorities assignment

In this case we define a process as a combination of node and traffic. To solve the PDP we need to provide mechanisms to handle starvation because a process may repeatedly enter the trying state after eating, possibly pre-empting the slots of its contenders. We use the doorway concept by Choy and Singh (1996).

The doorway algorithm works in such a way that if a process *p* finishes executing the doorway code, all neighbouring processes are blocked until *p* finishes eating. The doorway concept allows contending processes to set priorities based on arrival to the doorway. Any contenders who cross the doorway will have a higher priority (multiple processes may do so at the same time) over the process outside the doorway. Processes outside the doorway need to wait for those contending processes inside the doorway to finish eating. The main idea behind the doorway concept is that once any two contenders find each other outside the doorway, they do not need to check with each other again. Thus only when the contenders currently inside the doorway finish, they can enter the doorway to contend for the slot. The processes leaving the doorway need to check with all of its contenders before trying for the slot again. This guarantees freedom from starvation.

We use the doorway algorithm for every process *j*. A set $I_{(j)}$ keeps track of contenders that are inside the doorway, and a set $O_{\ (j)}$ keeps track of those outside the doorway. The algorithm is merged with the PDP algorithm of process *j* in order to obtain the full scale PDP algorithm. To facilitate the merge, we add an additional state, called *pending,* to the process states. After the thinking state and before moving to the trying state, a process enters the pending state in which it executes the doorway algorithm. If a process passes the pending state, it is inside the doorway and changes to the eating state.

Figure 61 depicts the DP problem that is reduced to one slot S_i, being contented by two nodes, node *1* and node *2*. I_1 and I_2 are the idle states of the nodes and B_1 and B_2 are the busy states of the nodes.

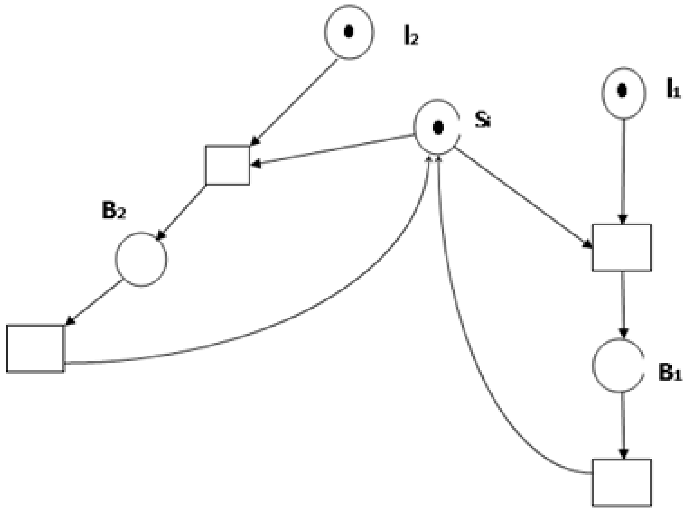


Figure 61: The DP problem for slot S_i, contended for by two nodes

The PDP algorithm is illustrated well by the Petri Net in Figure 62, which shows the transitions from one state to the other. In the diagram node *1* has higher priority than node *2*. However node *2* has access to slot S_i before node *1* and there are no more free time slots. Node 1 broadcasts a halt message that contains node 2's id. When node 2 receives the halt message it is forced to go into the Halt state and node *1* gets use of slot S_i and uses it for a certain time. When finished transmitting, node *1* releases the timeslot to node *2* for it to finish its transmitting.

Each node maintains a status table for all the time slots. When a node gains the right to use a slot it broadcasts to all its neighbours that the slot is in usage and all nodes change the status of the nodes from free (00) to busy (01) or pre-empted (10). When the node goes into the releasing state, it broadcasts that the slot is now free and all other nodes will update the status of the slot in their tables as shown in Table 27. When a node *x* pre-empts a lower priority node *y*, the slot status changes from busy to pre-empted. When node *x* releases the slot, node *y* resumes transmission and the slot status changes from pre-empted to busy. This makes it easy for nodes to identify free slots when they go into the trying state.

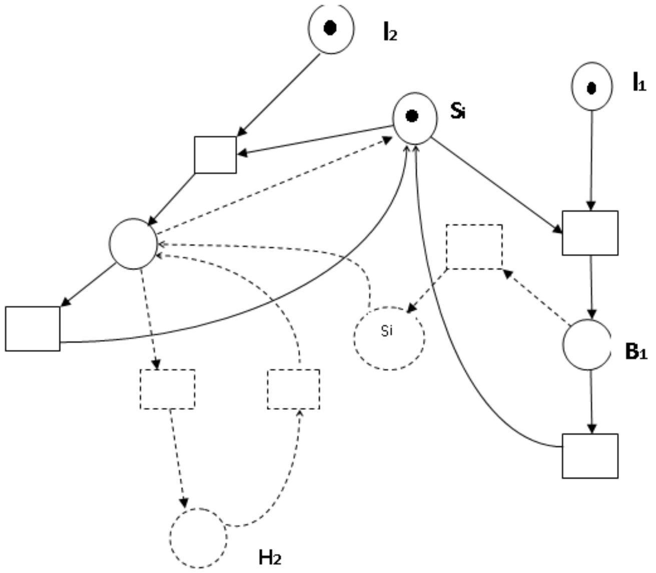


Figure 62: A Petri Net for the PDP model for two nodes in contention

Table 27: Slot status table maintained by every node

Slot number	Status	Node id	Priority
1	01	3	3
2	00	-	-
………….	…………………	……………….	…………….
………….	…………………	……………….	…………….
n	10	6	2

Prioritised Dining Philosophers Assignment (PDPA) Algorithm

Figure 63 is the flow diagram for the Prioritised Dining Philosophers Assignment (PDPA) Algorithm. It captures the various states of the node and the slot status. If a new flow requires access, the node checks for a free time slot.

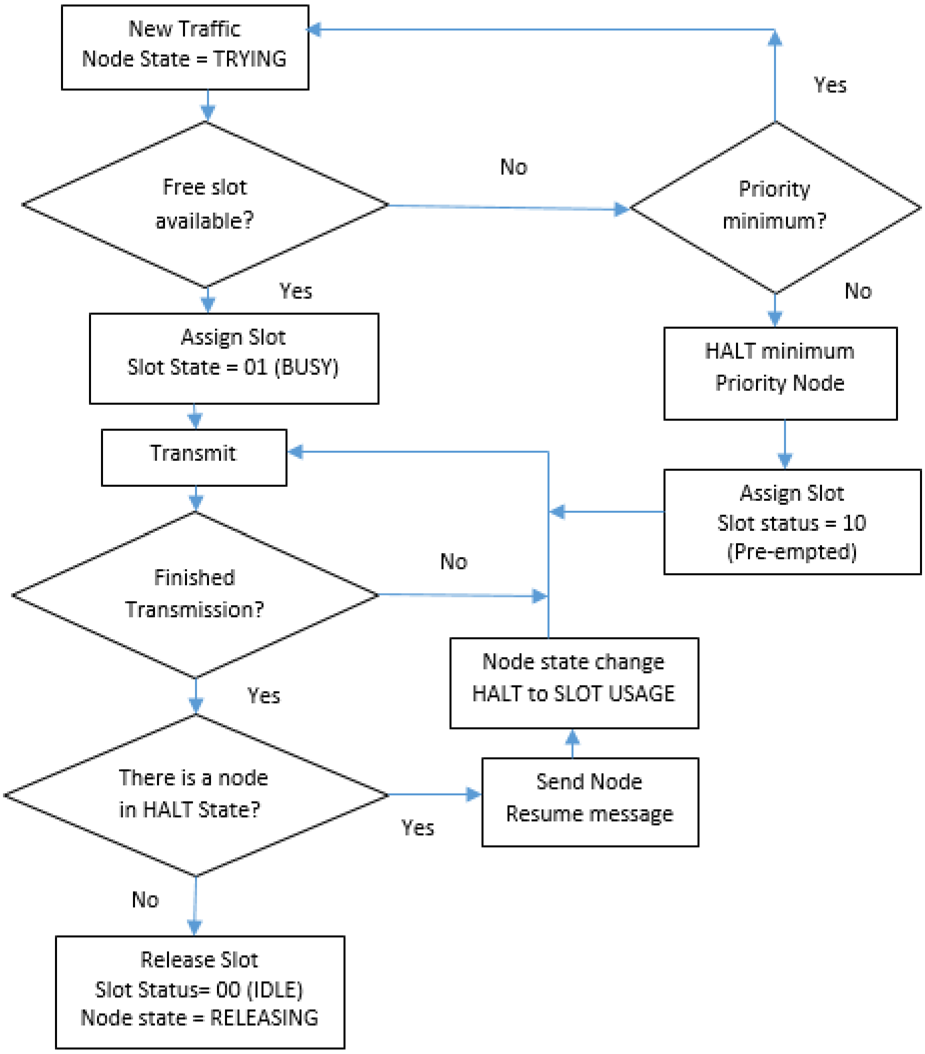


Figure 63: Prioritised Dining philosophers algorithm flow diagram

If a free slot is available, then it assigns itself that slot and update its status table, updating its neighbours at the same time. If there is not free slot, it checks if there is a lower priority node with slot assigned starting with the least priority. If it finds such a slot it gains use of that slot and broadcast a halt message containing the pre-empted node's id. All nodes update their slot status tables and the pre-empted node also change its status to pre-empted (10). Below we show the pseudo code for the prioritised Dining philosopher's assignment algorithm.

```
#INPUTS: A set of time slots S
#PROCESSES: TRYING, SLOT USAGE, HALT, REALISING
#OUTPUTS: Priority Based assignment of Slots
If (new traffic i) {
        Node (i) State = TRYING
        Node (i) check its table for free slots
        If (there is a free slot (x) )
                {Assign the slot (x) to the node (i)
                Broadcast assignment to all neighbours
                Neighbours update slot (x) state to 01        //BUSY state   }
        Else if (there is no free slot) {
        Embark on Priority Assignment ()}}
        Node (i) state = SLOT USAGE
        Node (i) transmit during the slot (x)
        If (When node (i) is finished)
                release slot (x)
                Node (i) state = Releasing
                Check if slot (x) state = 10    // another node (w) is in Halt state
                If (slot (x) state = 10){
                        Assign slot (x) to node (w)
                        Change slot state to 01         // slot state is Busy again }
                Else if (slot(x) state = 01)     // No node was pre-empted
                        Change slot(x) state to 00     // slot (x) is free
Priority Assignment () {
Check table for least priority traffic j assigned slot (y)
        If (least priority j exists) {
                Broadcast Halt () message with node (j) id.
```

```
            Node (j) state is HALT
            Assign time slot (y) to node (i)
            Slot (y) state =10              // Means slot was pre-empted
            Assign highest priority to node (j)   // to avoid starvation
            Node (i) state is SLOT USAGE
            Node (i) transmit in slot (y)   // use pre-empted time slot
            }
    }
```

5.4 Experimentation and Analysis

To model the performance of the PDPA algorithm we created a system comprising of a C++ code on which the simulations were done. Nodes were placed in space and flows were initiated at specific times of 0, 110, 240 and 350 seconds. Each time there are 10 flows initiated that want to contend for seven time slots. Table 28 shows a sample of the traffic characteristics of the network for one simulation. The traffic is shown as tuple, *Flow number, Size (Bytes)***.** In every time slot, a flow which is busy, transmits 30 bytes of data. Two simulation experiments were executed.

Table 28: Sample flows, priorities, sizes, and arrival times of traffic flows

Times/ seconds	priority1 Flow-Bytes	priority2 Flow-Bytes	priority3 Flow-Bytes	priority 4 Flow-Bytes	priority 5 Flow-Bytes
0	Flow7-210 Flow8–200	Flow4-300 Flow9–230	Flow2 -200 Flow6 -200 Flow10 -245	Flow3-215	Flow1-300 Flow5-300
110		Flow14–100	Flow11 -250 Flow17 -130	Flow13-300 Flow16-230 Flow18-200	Flow 12-300 Flow 15-200 Flow 19-230

				Flow20-340	
240	Flow23-200	Flow24-300 Flow 27-200	Flow21-300 Flow 22-200 Flow 26-300	Flow30-180	Flow25-300 Flow28-160 Flow29-145
350	Flow31-200	Flow37-300	Flow33-250 Flow38-330 Flow40-360	Flow34-270 Flow35-250 Flow39-340	Flow32-380 Flow36-300

The first experiment is the control experiment which does priority assignment without pre-empting. In the control experiment, traffic was assigned in such a way that higher priority traffic was assigned first before lower priority traffic. However the higher priority had no pre-empting capabilities such that if a lower priority traffic flow is being transmitted it is allowed to run its course and finish. The second experiment is for the PDPA algorithm. In the PDPA experiment higher priority flows were assigned before lower priority flows. When slots are used up, lower priority flows were halted and their slots were given to higher priority flows. The total times during which the flows of a specific priority have been blocked were calculated and recorded. The throughput for flows for specific priority flows was also calculated and recorded.

5.5 Results

The performance of the algorithm is measured in terms of the total time that traffic of a certain priority is waiting to access a time slot for transmission and the throughput of traffic of a certain priority, all against time. Many simulations were run and the average values of waiting time, throughput and number of admissions per type of traffic were calculated. Figure 64 and Figure 65 show the amount of time the traffic of each priority level, is waiting for transmission, for both the control and the PDPA algorithm.

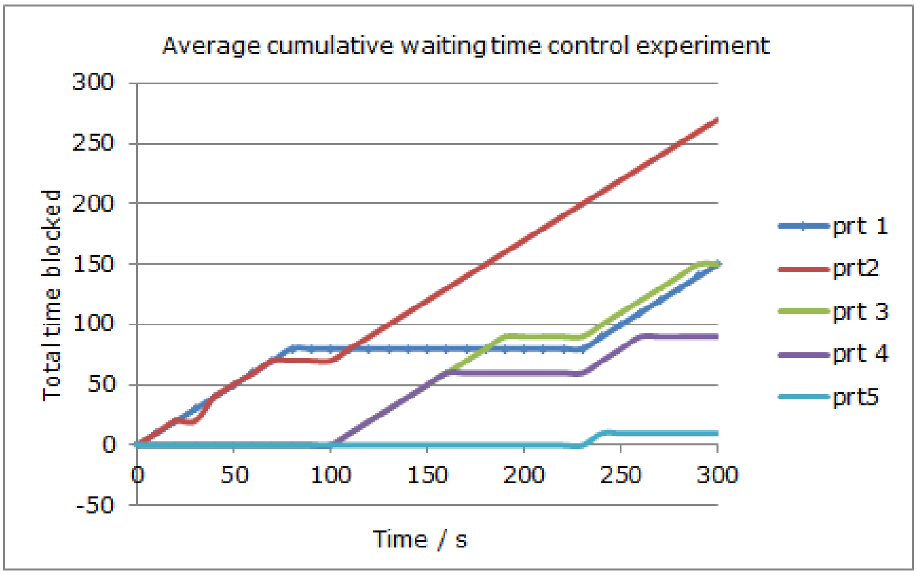


Figure 64: Average cumulative waiting time for the control experiment

In the control experiment, which relates to the priority assignment without pre-empting, the waiting time for traffic does not depend on its priority. It allows traffic of high priority, (4 and 5) to be blocked from transmitting more than in the PDPA algorithm. In the PDPA algorithm, high priority traffic is blocked from transmitting less than low priority traffic. Priority 5 traffic is not blocked at all except at the very end when it is blocked by traffic of its own kind. The lowest priority traffic, (priority1 traffic), is blocked more in the PDPA algorithm (Figure 65) than in the control experiment. However, priority 2 traffic is affected more in the control experiment than the PDPA algorithm. This shows that the PDPA algorithm affects lower priority traffic in such a way that it gives way to higher priority traffic compared to what happens in the priority assignment without pre-empting.

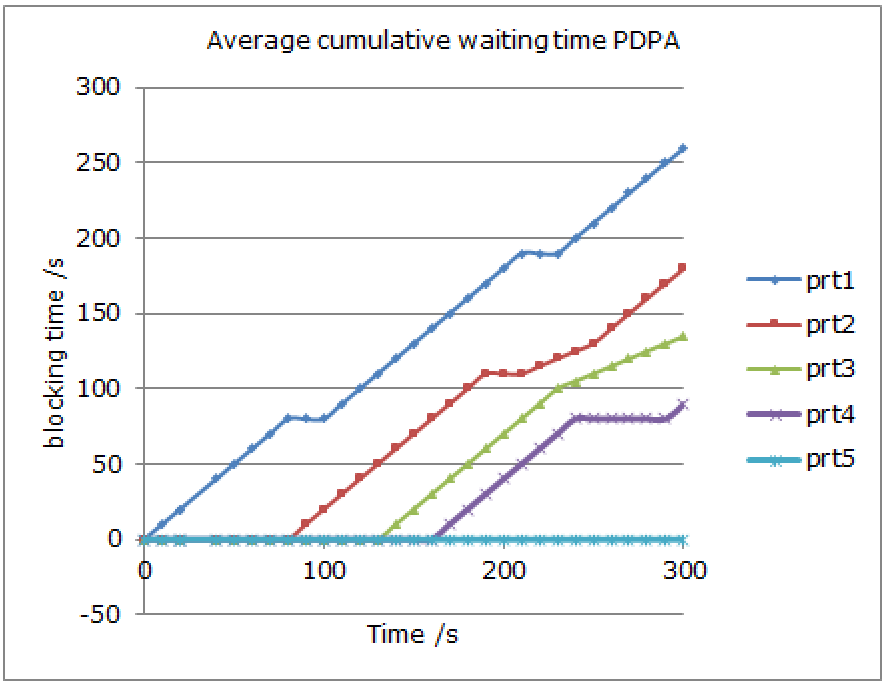


Figure 65: The waiting time for PDPA flows of various priorities

Figure 66 and Figure 67 show the variation of cumulative throughput for traffic of various priorities with time. Figure 66 is the control experiment that have not priority assignment and Figure 67 is the PDPA algorithm based experiment. Comparing the two experiments, we see that the throughput for higher priority flows, (priorities 4 and 5) is higher for the PDPA system than in the control experiment where the priority of traffic did not have an effect on the overall throughput. The throughput for the lower priority traffic (priorities 1, 2 and 3) is lower in the PDPA algorithm than in the control experiment. The highest priorities, (priorities 5 and 4) have high throughput until all their traffic is used up, that is when lower priority traffic (priorities 3, 2 and 1) start to increase in throughput as well. This shows that higher priority traffic will have more quality of service than lower priority traffic.

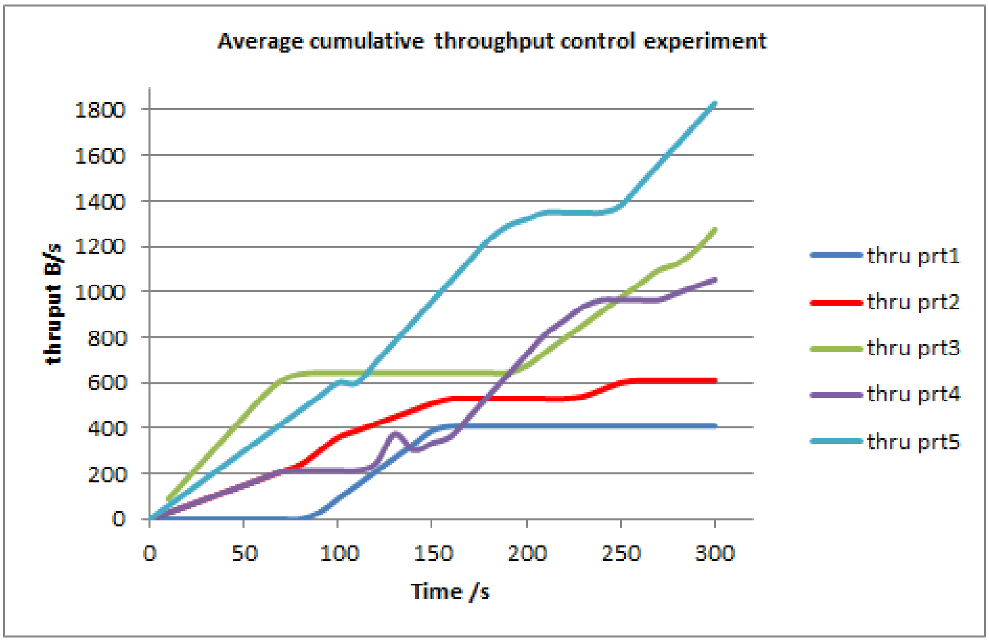


Figure 66: Throughput for flows for priority assignment for the control

In summary, the PDPA algorithm reduces the waiting time of higher priority traffic whilst increasing the waiting time of lower priority traffic. The PDPA also increases the throughput of high priority traffic and at the same time affects negatively the throughput of lower priority traffic. This happens because the high priority traffic is served first before all low priority traffic and new high priority traffic pre-empts low priority traffic and gets served first. The control experiment does not respect any priority so the throughput of the different traffic types does not follow any pattern.

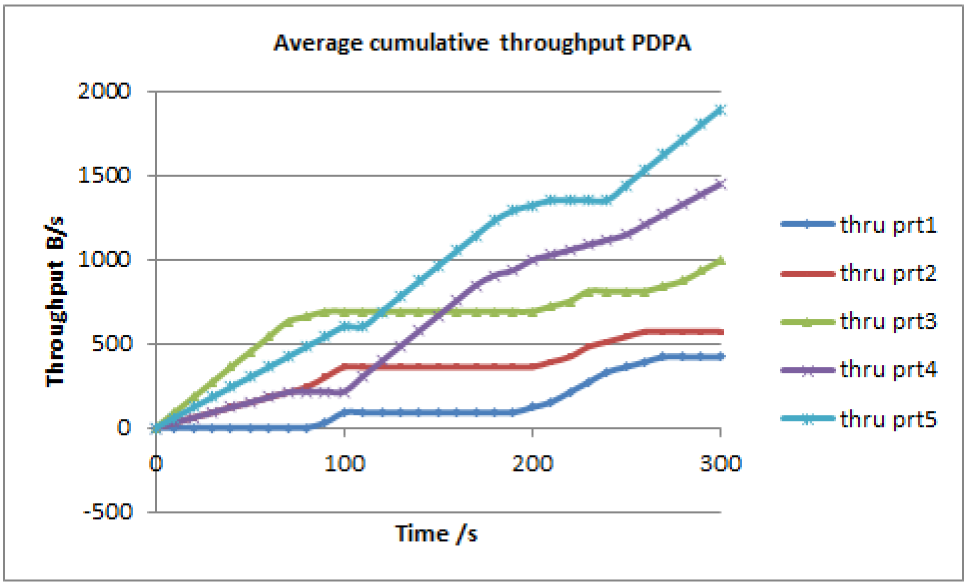


Figure 67: Throughput for PDPA flows for priority assignment

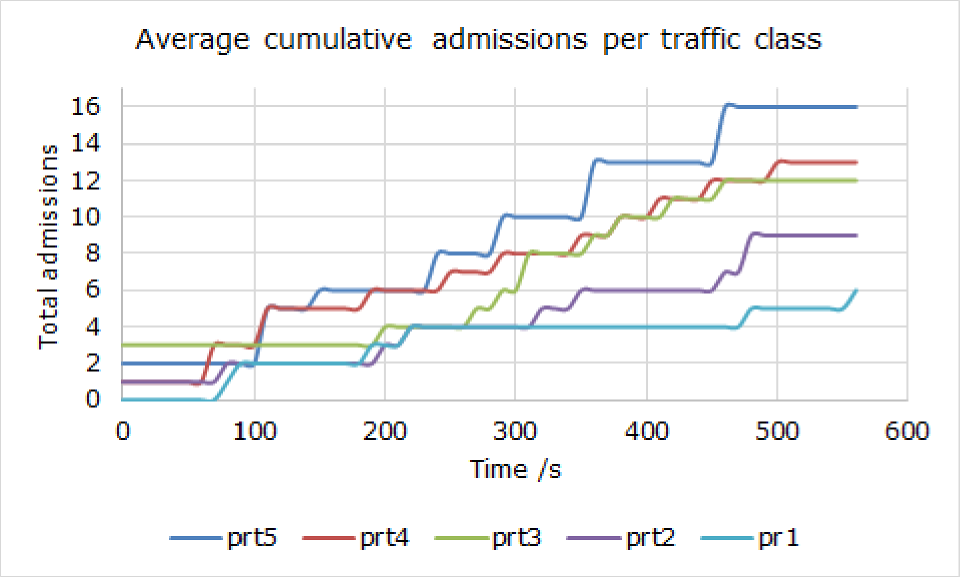


Figure 68: The variation of average cumulative admissions with time

The graph in Figure 68 shows the variation of total number of flows of each class admitted after a certain time duration. At the beginning there are three priority 3, two priority 5, one priority 4 and one priority 2 flows admitted. However priority 5 and priority 4 traffic admissions quickly increase than other lower priority flows. This happens because higher priority traffic is given pre-emptive priority in admission at the expense of lower traffic. Priority 1 and priority 2 traffic classes only increase slowly over time since they are admitted less regularly as compared to higher priority and they are continually pre-empted by higher priority traffic with time. This shows that higher priority traffic enjoys better quality of service than lower priority traffic. Although lower priority traffic, (priority 1 and priority 2) are pre-empted by higher priority traffic, they also increase their admissions with time they also increase in the number of admissions since the doorway algorithm prevents total blockage of such traffic.

5.6 Conclusion

In this chapter we came up with a time slot assignment method for TDMA based MANETs, that takes note of priority of traffic. Traffic was classified into various classes with varying priorities and the objective was to assign traffic to time slots in such a way that high priority traffic is assigned time slots ahead of low priority traffic. The main aim is to provide quality of service to traffic such that higher priority traffic have better quality of service. High priority traffic was given pre-emptive priority over low priority traffic in such a way that if high priority flow request assignment, it can pre-empt a low priority flow from its previously allocated time slot.

We modelled the problem as a Dining philosophers' problem and proposed a modification to come up with what we term prioritised Dining philosophers' algorithm (PDPA). Pre-empting results in low priority traffic are being blocked from transmitting for a long time and they run into risk of being starved forever. The doorway algorithm prevents starvation of lower priority traffic by elevating the priority of packets once they have been pre-empted once. The performance modelling of the PDPA proves that it successfully satisfies the initial objectives of this research of increasing the possibility of admission of higher priority traffic at the expense of lower priority traffic. It also succeeds in producing higher throughput for higher priority traffic. This means that higher priority traffic like VoIP and video over IP will have better chance of being admitted than lower priority traffic like HTTP data. The PDPA also succeeds in reducing the time that the highest priority traffic is blocked from transmitting thereby increasing the throughput of the same traffic. The algorithm increases the waiting time and at the same time reduces the throughput of lower priority traffic. The algorithm makes sure that the throughput for high priority traffic is high, so much that packets are transmitted early thereby improving quality of service.

In the future, it will be interesting to see the effect of allowing the lower priority traffic to be pre-empted more than once and come up with the optimum number of times the traffic can be put into halt state without starving the traffic. Also it will be very important to test the PDPA algorithm in a test bed so that we can test the effect of processing delay, queuing delay, transmission delay and propagation delay on the total performance of the PDPA algorithm. It would also be important to run the algorithm in a simulation environment like ns-2 and even in a test-bed scenario in order to measure how the algorithm compares to other algorithms by other researchers.

6. Security in MANETs[5]

6.1 Problem definition

MANETs have QoS vulnerabilities with respect to their QoS signalling methods. Because of their characteristics of open infrastructure, dynamic network topology, lack of central administration and limited battery-based energy of mobile nodes, MANETs are susceptible to both external and internal attacks.

MANETs have an open network topology. A node within transmission range of the MANET can join the network and its address and identity do not depend on its location. An attacker can join the network and because of the overlaps in radio range it can overhear QoS requests and control messages and can actively interfere with such messages. This makes the signalling protocol vulnerable to attacks on confidentiality and availability. Security enhancement schemes become worthless when the malicious nodes has already entered the network or some nodes in the network are compromised by the attacker. Such attacks are more dangerous as they are initiated from inside the network and because of this the first line of defence of the network becomes ineffective. Since internal attacks are performed by participating malicious nodes which behave well before they are compromised therefore it becomes very difficult to detect.

In a fixed and wired network, the IP address of a host is considered to be its identity and indicative of its location in a network topology. This is not possible in a MANET setting, because of mobility of nodes, so it is difficult to trace and verify the legitimacy of QoS requests. Due to intermittent connectivity, control messages may be lost or protocol timing dependencies may be modulated. Such effects are difficult to distinguish from real attacks.

[5] Work presented in this chapter is an extension of work published in the following journal article; Benny M. Nyambo, Willard Munyoka, G. K. Janssens and W. Lamotte. Achieving MANET Network-Layer Security through establishing Node and Route Trust. *International Journal on Information Technologies & Security*, № 4, 2009

6.2 System definition

For a reservation-based QoS signalling protocol like INSIGNIA, ASAP or AMAN many types of attacks can be executed by a malicious node once it is part of the network. These include over reservation, state table starvation, QoS degradation, replay and flooding attacks. (Zouridaki, Hejmo, Mark, Thomas, & Gaj, 2005) made a thorough analysis of the attacks that can occur in reservation based signalling protocols.

6.2.1 Over reservation attack

An over reservation attack occurs when a greedy node exploits the signalling protocol and reserve more bandwidth than what it actually needs to use for one of its real-time flows. The malicious node can also reserve bandwidth for non-existing flows in order to perform a denial of service (DoS) attack or to ensure that its own real-time applications could be supported in the near future. Most of reservation-based signal protocols cannot verify usage of reservations and they perform naive refreshment of reservations. To attack the network in such a case, the malicious node acts as the source node and requests more bandwidth than it uses. It then sends one data packet in the specified refresh-time interval to keep the reservation refreshed. The effect of such an attack is that bandwidth is under-utilized and legitimate sessions are denied service.

The smaller capacity of wireless network as compared to wired networks means that an over-reservation attack can create a DoS condition faster in MANETs than in wired networks. MANETs cannot use the straightforward techniques for rate monitoring due to the limited computational power of the mobile nodes. A solution that does not overwhelm the node capability should be sought.

6.2.2 State table starvation attack

The state table starvation attack is possible when the protocol requires flow reservations when a malicious node makes reservation of state for illegitimate flows and this leads to a state table exhaustion when the storage capacity of a node is exceeded. The vulnerability arises from the fact that a node has limited memory and computational power, reservations are made on a per flow basis and the reservation protocol cannot verify usage of reservations. In this mode of attack, the malicious node acts as the source of the data packets and requests bandwidth for an illegitimate real-time flow. The effect is that the state table is exhausted and legitimate sessions are denied service.

6.2.3 QoS Degradation

QoS degradation involves an increase in the delay or jitter of the real-time packets to unacceptable levels. This arises from the fact that the signalling protocol does not verify QoS performance. The malicious node acts as an intermediate node and increases the delay or jitter of the data packets to unacceptable levels. The effect is that QoS for a particular service is degraded and real-time session needs to be re-initiated. Increasing the delay or jitter of the real-time packets to unacceptable levels are attacks specific to real-time flows. Conventional DoS mitigation techniques cannot recognize the increase on delay or jitter of the real time packets. Thus, the current DoS-aware IDS schemes cannot defend the network against QoS degradation attacks in the Internet or wireless networks.

QoS degradation attacks are difficult to distinguish from normal degradation caused by the mobility of nodes or intermittent connectivity in the MANET. Monitoring QoS is a particularly difficult task for mobile devices in MANETs due to their limited capabilities.

6.2.4 Flooding attack

This is an attack when a malicious node floods the network with meaningless data packets which chew up the available bandwidth so that new flows are denied passage. Neither reservation-based nor reservation-less signalling protocols are resistant to flooding DoS attacks. The vulnerability arises from the fact that the protocol does not verify resource usage, does not identify the source of flooding and does not take measures against flooding. In this case the malicious node acts as the source node and floods the network with data traffic leaving the network flooded and legitimate sessions are denied service. One technique to mitigate flooding is to trace back the attacker and cut off the attack traffic at the source. However, it is much more challenging to trace back an attacker in MANETs than in a wired environment.

6.2.5 Replay Attack

Any protocol that allows the exchange of unauthenticated information is vulnerable to modification and replay. The replay attack can be performed by a compromised node on the route to the destination by duplicating and modifying the information in a signalling message.

Vulnerability arises from the fact that the protocol does not protect the integrity of signalling information, it cannot distinguish a replay from an authentic message and the topology is open each mobile node hears the transmission of every node in its radio transmission range. During the attack, the malicious node duplicates/modifies signalling information and forwards modified packet to the next hop. The effect is that resources are wasted by illegitimate packets and eventually legitimate packets are denied service.

Routing protocols play a very pivotal role in packet delivery and forwarding. Basically, there are three broad classes of routing protocols for MANETs: In *proactive/table driven protocols* every single network node maintains a routing table with updated information of accessible nodes throughout the network. Once the topology has changed, the routing tables have to be updated. *Reactive/On-demand protocols* establish their routes between nodes only when they need to route or forward data packets. There is no updating of every possible route in the network and within each mobile node, but instead, it focuses on routes that are being used or being set up. The routing process is divided into route discovery and route maintenance processes. *Hybrid protocols* combine the strengths of both proactive and reactive routing protocols in finding efficient and trusted routes (Corson, Macker, & Cirincione, 1999).

Both proactive and reactive routing protocols assume a general trustworthy and cooperation of participating nodes i.e. adopts an a priori trust (Mojdeh, 2003).

This general assumption and the intrinsic nature of MANETs make routing protocols vulnerable to routing disruption attacks leading to Denial of Services (DoS) attacks (Deshpade, 2004), (Zapta, 2002).

In this chapter we aim to come up with a set of efficient, trusted routing discovery and maintenance rules, and a security framework for establishing trust of cooperating nodes. This should make use of neighbour verification and monitoring techniques to enhance the Network-Layer security. This is a hybrid security solution that assumes no *a priori* trust between network nodes and makes use of both first-line preventative mechanisms like authentication and encryption; and second-line defensive mechanisms like intrusion detection and reaction systems.

Our approach adopts a probing technique in which every legitimate network node carries a token with a specified period of validity assigned with a secret key and this key can be verified by its neighbours. Before this token expires, each network node must renew its token from its neighbours, who in turn collaboratively monitor it to detect any misbehaviour. In their operation, each node promiscuously listens to the activities of its neighbouring nodes to cross-check if any one of them performs route cache poisoning, has failed to perform packet routing and forwarding functions, or if it is constantly sending/ forwarding corrupted packets. We will also design a hybrid node and message authentication algorithm using the combined information from our proposed trust framework and trusted routing discovery and maintenance set of rules. When fully integrated, these features constitute a host based intrusion detection system for a secure reactive routing protocol.

6.3 Model formulation

Our fundamental goal in this security framework is to provide a coherent and unified Network-Layer Security solution to protect both the routing and packet forwarding functionalities in the DSR routing protocol for mobile ad hoc network. It is paramount at this point to explicitly distinguish the vulnerabilities of the Network-Layer as packet routing misbehaviour and packet forwarding misbehaviour. The difference lies in that the routing functionality is only responsible for establishing and maintaining the routes; but it cannot enforce that the data packets are correctly forwarded along the routes. Since we seek to come up with a robust security solution to these two key misbehaviours, we first elaborate on each one of them:

i. **Routing update misbehaviour** – this pertains to any action of advertising routing updates that does not follow our routing discovery and maintenance rules. Our solution seeks to curb the following negative malicious hacker actions (Marti, Giuli, Lai, & Baker, 2000):

   - Advertising a route with smaller distance metric than its actual distance to the destination.
   - Advertising routing updates with larger sequence numbers and invalidate all the routing updates from other network nodes.
   - Spoofing of node IP address and advertising a falsified broken operational link.
   - Forwarding packets along non-optimal routes, non-existent or with poor quality or on a worst case, may intentionally introduce severe network congestion and network contention in certain areas or partition the network, create routing loops and waste network resources like bandwidth and battery power.

ii. **Packet forwarding misbehaviour** – this pertains to any malfunctioning of the data packet forwarding services as a consequence of a malicious attack. These also form a core area to our proposed solution and we focus on three types(Marti et al., 2000) (Capkin, 2003):

   - Selective packet dropping – here a malicious node systematically drops data-packets that it is supposed to forward to its neighbours.
   - Packet duplication – here a malicious node replays old data packets that it has already forwarded.
   - Network-Layer packet jamming – here a misbehaving node sends too many data packets into the network and occupies a significant portion of the bandwidth.

Although we have differentiated the two misbehaviour functionalities for clarity's sake, we will come up with a unified solution. In our proposed security framework we adopt a token-based security system, in which every legitimate network node carries with it a token signed with a system secret key which can be verified by its neighbours.

From the stated misbehaviours we deduce that our security framework should achieve four key goals for its smooth functioning, which are:

i. Our security framework must tolerate the coexistence of compromised nodes. Here we treat a compromised node and a hacker as one. However, to achieve a workable solution, we use unique ID (IEEE 802.11 MAC address) for unique node identification and exchange symmetric keys and encrypt data packets to achieve data confidentiality and integrity (Levien, 1998), (Capkin, 2003), (Kranakis, 2003), (Marti, 2000).

ii. The security solution should be self-organized with no dependence upon any centralized trusted entity like key distribution centre (KDC) for establishing trust relationships between different nodes. We do not assume any 'a priori' secret association or trust relationship between nodes.

iii. The security solution should proactively isolate malicious nodes from the network and this guarantees the elimination of denial of service (DoS) attacks in the Network-Layer.

iv. Finally, our security solution should have a decreased overhead and increased trustworthiness over time with operation and this suffices the resource-constrained/reservation requirement of MANETs.

Our security solution exploits great collaboration among local nodes without completely trusting any individual node – the reason being that an initially authenticated legitimate node can perform Byzantine effects (Ngai, 2004), (Levien, 1999).

Figure 69 illustrates the composition of our proposed network-layer security solution, which constitutes four closely interacting components: Secure path finding module, trust management module, neighbour verification module and intrusion reaction/response module.

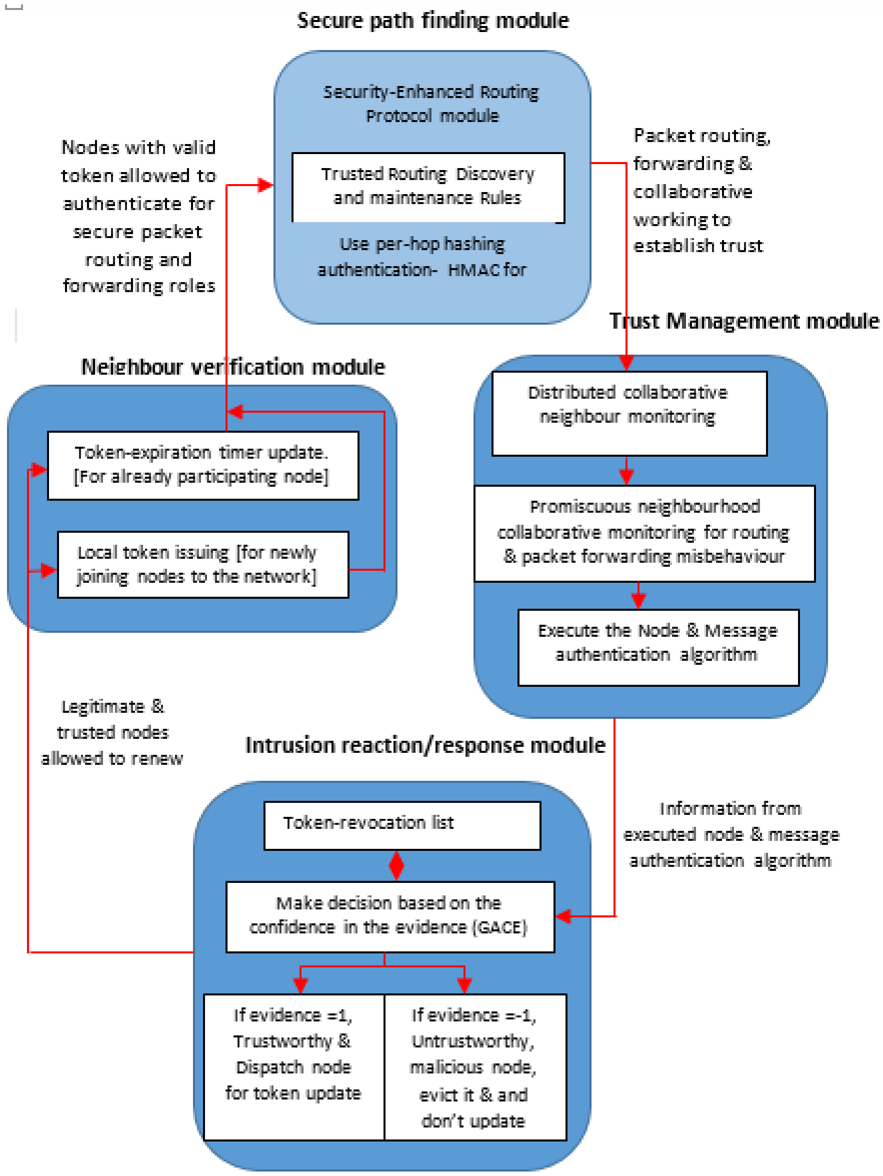

Secure path finding module
Security-Enhanced Routing Protocol module
Trusted Routing Discovery and maintenance Rules
Use per-hop hashing authentication- HMAC for
Nodes with valid token allowed to authenticate for secure packet routing and forwarding roles
Packet routing, forwarding & collaborative working to establish trust
Trust Management module
Neighbour verification module
Token-expiration timer update. [For already participating node]
Local token issuing [for newly joining nodes to the network]
Distributed collaborative neighbour monitoring
Promiscuous neighbourhood collaborative monitoring for routing & packet forwarding misbehaviour
Execute the Node & Message authentication algorithm
Legitimate & trusted nodes allowed to renew
Intrusion reaction/response module
Information from executed node & message authentication algorithm
Token-revocation list
Make decision based on the confidence in the evidence (GACE)
If evidence =1, Trustworthy & Dispatch node for token update
If evidence =-1, Untrustworthy, malicious node, evict it & and don't update

Figure 69: Composition of the proposed network layer security solution

6.3.1 Neighbour Verification

Here we base our mechanism on tokens and key management since these are popular methods in securing networks (Pourmir, 2014) (Joshi, Srivastava, &

Singh, 2010). It employs the asymmetric cryptographic primitives i.e. the RSA. We define a global secrete-key-pair of SK_i/PK_i, in which PK_i is made known to all nodes joining the mobile ad hoc network. Each legitimate node carries a token stamped with an expiration-time and should be signed by SK_i. For easy and proper function, our token has five key-fields namely: specific-identity (IEEE 802.11 MAC address), signing-time, expiration-time, one-hop count and sequence-number.

Each node should periodically broadcast its token in a 'hello' message to its neighbours for verification and token update. Each token is regarded as valid if and only if:

- it is signed by SK_i,
- it has not expired, and
- it is held by the node with the same identity as stated in its specific-identity field.

We regard any node without a valid token as a malicious node and all its subsequent data packets and routing updates must be dropped by its neighbours.

In our design we assume a decentralized token issuing process in which every node actively issues tokens to its neighbours. This guarantees us fault-tolerance IDS with independent failures and isolation of malicious nodes.

6.3.2 Localized Token Issuing Process

The issuing of tokens amongst neighbouring nodes is two-fold as illustrated in Figure 70.

a) Existing node needing to renew its current token

We model our message handshaking in the localized token issuing process as an undirected graph *G*, where *G* = (*V, E*) – consisting a set of *n* nodes (vertices) and a set of *m* node pairs (edges). The set of nodes, denoted by *V* = {1, 2 … *n*}, represent network–enabled ad hoc devices and the set of edges, denoted by ***E*,** represent the wireless communication links. The topology of graph *G* and the set of node pairs are dynamic; and we define node *Z's* neighbours in Figure 70 as those nodes that can update its token and are in dark colours i.e. nodes 1 up to

4. Nodes 5 up to 9 are within two-hop neighbourhood to *Z*. Thus they do not have a direct communication link with node *Z*.

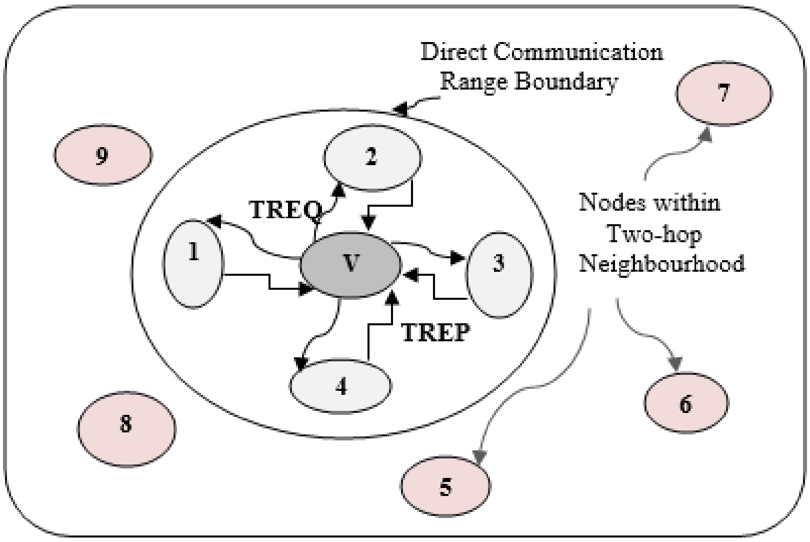

Direct Communication Range Boundary
7
9
2
TREQ
1
V
3
Nodes within Two-hop Neighbourhood
TREP
4
8
6
5

Figure 70: Localized token issuing and message handshaking

Before node *Z's* current token expires, it must broadcast a TREQ (Token Request) packet to its neighbours. This token request contains its specific-identity and timestamp. Each node should keep a Token Revocation List (TRL) learnt from our intrusion reaction and response module and use this to make decisions on whether or not to serve the token request from *Z*.

When a node receives a TREQ packet from its neighbour, it extracts the token, checks whether the TREQ comes from whom it claims to be and whether the token has already been revoked by consulting the TRL. If the token is still valid and the source of the TREQ is verified, the neighbour must construct a new token in which specific-identity matches the old token, signing-time is equal to the timestamp in the TREQ packet and expiration time is determined by the additive increase algorithm given below. The neighbour should then sign the newly constructed token using its individual SK_i, encapsulate the partially signed token in a token reply (TREP) packet and then unicast it back to node *N*. In our model, those token request packets of which the sources cannot be verified or containing already revoked tokens are silently dropped and cannot participate again in the packet routing and forwarding without a valid and updated token.

b) Newly joining nodes requiring their first tokens

This is almost similar to the token renewing scenario in the message handshaking primitive. To join the network, a new node has to broadcast a TREQ packet containing its specific-identity and the current-time to its neighbouring nodes. We adopt the same procedure of issuing a TREQ here, as that applied by a current node renewing its token. However, the only difference lies in the expiration-time fields. A newly joining node has a smaller period of validity because of the uncertainty factor and minimal trustworthiness attached to it, whereas a renewing node has larger expiration-times.

c) Determining period of validity of each token

We define the expiration-timer of a token (expiration-time field) as the trade-off between the computational overhead and the length of token revocation list (TRL). Choosing a larger expiration timer will decrease the computation overhead as fewer token renewal processes are required. However, it will also increase the expected length of TRL because once the token is revoked, it will remain in the TRL for more than 10 minutes before it expires.

We adopt a credit-based strategy in determining the expiration timer of each node’s token. The period of validity of a node’s current token is dependent on how long it has stayed and behaved well in the network. A newly joined node is issued a token with small period of validity. When it keeps on behaving well in the network, its subsequent tokens will have longer and longer period of validity. This is achieved by additively increasing the period of validity when a node renews its token from its neighbours.

Let ST_1, ST_2, ST_3, and ST_4 denote the signing-time and expiration-time fields in the previous and renewed tokens, respectively. The additive increase algorithm states that:

$$ST_4 - ST_3 = ST_2 - ST_1 + ST_0 \qquad 6.1$$

In the credit based strategy, when a node receives its n^{th} token, the duration of the time it has stayed in the network can be calculated as:

$$ST_L = \sum_{i=1}^{n-1} i * ST_0 = \frac{n(n-1) * ST_0}{2} \qquad 6.2$$

6.3.3 Trust manager and neighbour monitoring

In the trust manager and neighbour monitoring module, each network node promiscuously monitors the behaviour of its neighbours and detects any malicious behaviour in both routing and packet forwarding services. It is within this module that our proposed hybrid node and message authentication algorithm is embedded. We regard any detected misbehaviour as evidence of malicious attacks.

a) Monitoring routing update misbehaviour

Routing update misbehaviour is detected by examining the correctness of routing updates. This greatly relies on the distributed collaborative monitoring and promiscuous listening mechanisms to increase mobility accuracy and withstand routing disruption attacks. On receipt of a route reply (RREP) packet broadcast by its verified and legitimate neighbours, the receiving one-hop node has to examine the correctness of the newly offered route. All nodes within one-hop neighbourhood compare the new route entry with its cached route entry previously announced. We regard the new route entry correct if and only if the sequence number in the two route entries are identical and the one-hop count in the new route entry is one larger than the hop-count in the cached route entry announced by the promiscuously monitored node. Once the monitoring node discovers routing update anomalies, it must drop the RREP packet and broadcast a single intrusion detection (SID) alert message packet to its neighbours.

b) Monitoring packet forwarding misbehaviour

This is achieved in MANETs through overhearing the channel in promiscuous mode in the IEEE 802.11 Link-Layer. To address the packet forwarding misbehaviour, we adopt and improve the Watchdog in which after sending a packet, node X has to overhear in promiscuous mode, the forwarding of the data packet to the next correct one-hop neighbour – failure to do so within "drop-time" seconds, it considers the packet to have been dropped. If the bandwidth corresponding to the packets dropped by its one-hop neighbour exceeds a pre-defined threshold "duplicate-bandwidth", that node should be blacklisted as malicious and the SID packet should be broadcast to notify all the other network nodes. In our proposed node and message authentication algorithm, we utilize the information obtained by overhearing the channel to detect packet duplication and packet jamming. If one node overhears that the bandwidth corresponding to the duplicate forwarding of packets by its one-hop neighbour exceeds a predefined threshold "duplicate-bandwidth" or the bandwidth corresponding to packets-sent exceeds the threshold "Sending-bandwidth", then this should be considered as a malicious attack and should broadcast an SID packet to notify all the other network nodes and update the TRLs.

c) Distributed Collaborative Monitoring

In order to improve the monitoring accuracy and to withstand routing disruption attacks, we propose the use of "*m* out of *N*" strategy to cross-validate the monitoring results of different nodes in their one-hop neighbourhood. A node is considered as an attacker if and only if *m* nodes out of all its *N* neighbours have independently sent out SID packets against it. The "*m* out of *N*" strategy can significantly improve the accuracy of monitoring results and minimize the false positive rates associated with MANET intrusion detection systems. Two types of errors can be defined: Class I error which refers to a failure to detect the attacker and a Class II error which refers to false accusation against a legitimate node.

Let P_1 and P_2 denote the probabilities of a Class I error and a Class II error in the monitoring results made by a single node, respectively. By this collaborative monitoring, the detection probability for an attacker is:

$$P_D = \sum_{k=m}^{N} \frac{N!}{k!(n-k)!} (1-P_1)^k P_1^{\,N-k} \qquad 6.3$$

$$P_F = \sum_{k=m}^{N} \frac{N!}{k!(n-k)!} (1-P_2)^k P_2^{\,N-k} \qquad 6.4$$

where *m* is the number of one-hop nodes that has indicted a malicious node by broadcasting a SID,
N is the monitored node's one-hop neighbours,
P_D is the detection probability for a malicious node,
P_F is the false detection probability for legitimate node,
k is the secrete sharing parameter,
$(1\text{-}P_1)^k\ P_1^{\ N\text{-}k}$ denotes the probability of a node's failure to detect an attacker and
$(1\text{-}P_2)^k\ P_2^{\ N\text{-}k}$ denotes the probability of a node's false accusation against legitimate nodes.

To implement the collaborative monitoring mechanism, we also make use of the polynomial secret sharing scheme. However, it should be noted that in this chapter we do not differentiate the SID packets triggered by the routing updates misbehaviour and the packet forwarding misbehaviour. When a node has received *m* independent SID packets against the same node, it constructs a notification of token revocation, signs the notification using its own share of SK_i key**,** encapsulates the signed notification in a group intrusion detection (GID) packet and then broadcasts the GID packet as an alarm message. The first node that receives *k* GID packets against the same node combines them and constructs a token revocation (TREV) packet which is signed by the SK_i key**,** based on the polynomial secret sharing cryptography primitives and broadcasts it to the entire wireless network. The malicious node is silently evicted by not renewing its token and blacklisted in the TRLs so that when it tries to re-join the network, its specific-identity will be kept and is barred henceforth.

Here we present the node and message authentication algorithm for Secure and Trusted Routing. This algorithm was constructed using all the information provided in our proposed Network-Layer Security Solution and it authenticates both the network nodes and all the routed and forwarded messages.

Secure and Trusted Routing algorithm

```
IF (there is data to be routed to destination){
    IF (the route-entry path exists in its route-cache){
    Create the data-packet
    and send it to its destination
    And wait for the Link-Layer Ack }
    ELSE {
        Create the data-packet and encrypt it with the shared-key, PKi
        }
    IF (supporting-nodes in one-hop neighbourhood) {
        For (all n nodes in one-hop neighbourhood) {
            IF (rn, cn, bn) > (rreq, creq, breq) // assess for (trust) {
                Broadcast to neighbour n,
                    }
                }
        }
    }
On receipt of RREQ data-packet
Check the validity of data-packet
IF (data-packet maliciously altered and need validation) {
    Ignore forwarding the RREQ data packet and do not renew its token
    Broadcast an SID packet to neighbours a notifying all the TRLs
    }
 ELSE IF (Data-packet is correct & valid){
        Append own IEEE 802.11 MAC Addresses to its route-entry-
header,
        Renew token,
        Forward data-packet to two-hop neighbourhood node
        Perform passive –Ack
        }
IF (data-packet is dropped or corrupted){
    Broadcast a SID packet to neighbours
    Send data-packets to last supporting node
    Wait for ACK message for a period // time-out duration, 20 seconds
    }
IF (no ACK/-ACK received){
    Execute INTRUSION
    Aggregate all generated alerts in GACE}
IF (confidence of evidence = -1){
    Blacklist malicious node
    Evict it from network and
    Send an alarm message to all nodes
     Record its details in TRL
    }
ELSE IF (confidence of evidence = +1) {
    Node is legitimate but mistakenly labelled malicious;
    Renew its token expiration-time,
    Record its specific-identity in TRL
    Promiscuously monitor it for possible misbehaving actions
    }
Exit // +ACK received and data-packet successfully transmitted to destination
```

6.3.4 Secure Path Finding

a) Route Discovery Procedures

Here the source node S**,** intending to communicate with a destination node D, checks if D is within its one-hop count. If it is not, the source node has to generate a data packet, to encrypt it with a shared secrete-key SK_i (using our proposed Trusted Routing Discovery Rules) and to dispatch it. Only legitimate one-hop neighbour nodes, with the knowledge of the SK_i, can decrypt the data packet and process it. Our approach works by searching for either the destination or supporting node in the one-hop neighbourhood of S. If it fails, the encrypted route packet is selectively broadcast, considering three qualities of services (QoS) metrics namely: reliability/trust (R_{ik}), network congestion (C_{ik}), and bandwidth (B_{ik}) as follows:

$$R_{\max} = \max_{k \in \{1,2,\ldots r\}} \{R_{ik}\} \qquad 6.5$$

$$C_{\min} = \min_{k \in \{1,2,\ldots r\}} \{C_{ik}\} \qquad 6.6$$

$$B_{\max} = \max_{k \in \{1,2,\ldots r\}} \{B_{ik}\} \qquad 6.7$$

where, $k \in \{1, 2, \ldots, r\}$ is the neighbour of source S_i and R_{max}, C_{min}, B_{max} are the boundary parameters for trust/reliability, congestion and bandwidth respectively. S_i must set a threshold for each of these three parameters and selectively broadcasts data packets to its one-hop neighbours, meeting or surpassing this threshold. Reliability/trust of a neighbouring node is determined by taking into consideration its previous forwarding behaviour and reputation in the TRLs. The congestion parameter can be determined by communicating current load conditions among neighbours. The bandwidth is determined by communicating the current load on each link and taking into consideration bandwidth allocation time-slots.

b) Trusted Routing Discovery Rules

We present below our proposed trusted routing discovery rules. Our routing discovery rules are closely similar to those of Ariadne protocol (Hu, Perrig, & Johnson, 2005), however we incorporate additional security features to come up with our security-enhanced trusted routing discovery rules, which we call DSR-S*. Each node along the path to the destination D has to verify the origin of the RREQ data-packet using valid-token information about previous one-hop neighbour suggested in the neighbour verification module. Authenticating data alone in routing messages is not enough, because a malicious node can simply remove a node in the RREQ leading to lost/misrouted packets. Thus, we use a per-hop; two-way hash function utilizing message authentication code (MAC) computed with keys *SK_{iD}* (for RREQ from source node *S*) and *SK_{iS}* (for RREP from destination node *D*) over unique data i.e. signing and expiration times and one-hop count. This guarantees no omission in hop counts made during RREQ and RREP. To address the problem of Byzantine nodes participating in the packet routing and forwarding process, we adopt a collaborative working with the Trust Manager and the Neighbour Monitoring module for reputation/trust of a node in its previous participations i.e. whether it has been blacklisted before or not. Our Route Discovery rules are shown in the algorithm below, where the initiator node *S* is attempting to discover a route to the target node *D*.

S : *$h_0 = MAC_{SKiD}$ [RREQ, S→D, S.I,*
T.S, H.C, Seq#]

S* : *RREQ, S→D, S.I, T.S, H.C, Seq#,*
h0, ()

B : *$h_1 = H$ [B, h_0]*
$M_B = MAC_{SKB,\ T.S}$ [RREQ, S→D,
S.I, T.S, H.C, Seq#, h_1, B ()]

B* : *[RREQ, S→D, S.I, T.S, H.C, Seq#,*
h_1, (B), M_B]

C : *$h_2 = H$ [C, h_1]*
$M_C = MAC_{SKC,\ T.S}$ [RREQ, S→D,
S.I, T.S, H.C, Seq#, h_2, (B, C), M_B]

C* : *[RREQ, S→D, S.I, T.S, H.C, Seq#,*
h_2, (B, C), (M_B, M_C)]

D : *$M_D = MAC_{SKTS}$ [RREQ, S→D, S.I,*
T.S, H.C, Seq#, (B, C), (M_B, M_C)]

D→C : *RREP, D→S, T.S, (B, C), (M_B, M_C),*
$M_{D'}$( )

C→B : *RREP, D→S, T.S, (B, C), (M_B, M_C),*
$M_{D'}$, ($SK_{C,\ T.S}$)

B→S : *RREP, D→S, T.S, (B, C), (M_B, M_C),*
$M_{D'}$, ($SK_{C,\ T.S}$, $SK_{B,\ T.S}$)

where, *S* – is the source node, *D* is the destination node, *S.I* is the Specific-Identity (IEEE MAC Address) for both the *S* and *D*, *H.C* is the Hop-Count, *T.S* is the Time Stamp e.g. signing and expiring time, SK_B and $SK_{C\text{ is}}$ the secret MAC keys shared between nodes *B* and *C* (one key for each communication direction) and $MAC_{SKC,T.S}(M_C)$ denotes the computation of MAC of message M_C using MAC-key, $SK_{C,\ T.S}$.

c) Trusted Routing Maintenance Rules

Each of the forwarding nodes performs route maintenance to discover problems with each selected route. We design it in such a way that each network node, starting at a node where communication stopped backs to the immediate node just before the source node, should authenticate/certify the PKED-RERR. It should append a timestamp at which communication breakdown occurred. We propose the use of a piggybacked negative acknowledgement (PKED-RERR) to the source, rather than just an ordinary Route-Error message – this should be signed by the node at which the break occurred and forward it back until it reaches the source node. On receipt of the PKED-RERR message, the source node must verify the authenticity of the node at which the path has broken using its TRL information. If it has been once blacklisted before, this could be a possible attacker and the source node should broadcast a SID packet notifying other network nodes. In trusted route maintenance rules below, node *C* determines that its next-hop to node *D* is unreachable, it creates and signs a piggybacked-route-error message and forwards it back to node *B*, who also has to certify it before forwarding it to the source node S.

C→ B : $[(PKED\text{-}RERR, S \rightarrow D, Cert_C, T.S,$
$(B, C))_{SK\hat{C}}]$
B→ S : $[(PKED\text{-}RERR, S \rightarrow D, Cert_C, T.S,$
$(B, C))_{SKB}]$

d) Intrusion Reaction/Response

Our proposed intrusion reaction mechanism guarantees that the malicious intruder is isolated in the network once it is detected by its neighbours and it will never be issued with a new token again in the future. The "Execute Intrusion" sub-module of our node and message authentication algorithm can be embedded in this intrusion reaction module and collaborate with the Global Aggregation Correlation Engine (GACE) to make sound intrusion detections. We view this to be a global reaction scheme, in which alert aggregation and correlation takes place to generate global-alarms or global intrusion packet (GIP). In a zone based intrusion detection system (ZBIDS) this is performed by inter-zone; however, in our case all nodes must actively participate in this process.

There is a great collaboration between the intrusion reaction module and the trust manager and neighbour verification modules. For it to be properly functional and well-coordinated, each node that receives a Token Revocation (TREV) packet must verify whether the packet is signed by SK_i and whether the revoked token is already on the TRL. All TREV packets that are not signed by the secret-key SK_i, are regarded to have emanated from Byzantine nodes and must be silently dropped as they are regarded as malicious. All those first blacklisted TREV packets must be rebroadcast for further verification and authentication by its neighbours. Once there is strong evidence that a node is identified as malicious by its legitimate neighbours – all links between them must be broken using the path maintenance mechanism/ non-renewal of its token. The fact that there are no traffic concentration points in ad hoc networks, each of our localized intrusion detection mechanism promiscuously monitors its neighbour to detect any packet routing and forwarding misbehaviour.

We also adopt a reputation-based trust approach to make our decisions on the identified and blacklisted nodes. It is a combination of formulas that encompasses direct and recommended trust values. Thus, trust in a node is associated with a reputation value *T*, which is the same as the evidence – this represents the trustworthiness of a node. A node *X* in the network is considered by another node Y either as:

- Trustworthy; for $T = 1$, *if and only if* $R_t < R < R_{max}$
- Untrustworthy; for $T = -1$, if and only if $R_{min} < R < R_u$

where *R* represents the reputation value of node *X/Y* and trust value T in the range: $R_{min} < R < R_{max}$, with two threshold: $R_u > R_{min}$ for untrustworthy and $R_t < R_{max}$ for trustworthy.

We also propose and introduce a fade factor '*w'* to give less weight to evidence received in the long-past to allow for reputation erasure in the TRL.

6.4 Experimentation and analysis

In order to measure as to how much our hybrid security solution performs in improving Network Layer security, we simulated our proposed security solution using a C++ code. The simulation parameters are shown in Table 29. For each data point, a huge number of simulations (each time with different seeds) were run randomly to obtain average values. We used the medium access control (MAC) layer and transmission control protocol (TCP) data traffic for our simulation to evaluate the performance or our security solution.

Table 29: Parameter values

PARAMETER	**VALUE**
Space	700m × 700m
Number of Nodes	26
Mobility Model	Random Waypoint
Maximum Speed	5-20m/s
Traffic Type	10 CBR Connections
Pause Time	60 seconds
Packet Generation Rate	1-20 packets per second
Packet Size	512 bytes
Total simulation Time	1507 Seconds
Wireless Transmission Range of each node	250m

6.4.1 Performance Metrics

The following performance metrics are used throughout the simulation to evaluate the performance of our hybrid IDS scheme with respect to TCP traffic.

a) Packet Delivery Ratio (PDR):

This is the ratio of number of data packets generated by CBR source nodes that are successfully delivered to the destination nodes divided by the number of data packets transmitted by the source node. It evaluates the effectiveness of DSR-S* to deliver data packets to their destinations in the presence of malicious agents which selectively drops packets they are required to forward. The packet delivery ratio is directly influenced by packet loss, which may be caused by incorporative behaviour. In our case, this is also referred to as Route Trust – thus, measures the reliability that packets will reach their destination if forwarded on a particular route. It is calculated as follows:

$$PDR = \frac{Number\ of\ packets\ received\ at\ destination}{Number\ of\ packets\ forwarded\ by\ the\ node} * 100 \qquad 6.8$$

b) False Positive Ratio (FPR):

This is the percentage of decisions in which legitimate network nodes are mistakenly flagged as malicious or misbehaving nodes, or failure to detect the actual malicious nodes.

$$FPR = \frac{Number\ of\ misclassified\ nodes}{Number\ of\ malicious\ nodes\ detected} * 100 \qquad 6.9$$

c) Malicious Detection Rate (DR):

This measures how well our proposed DSR-S* with IDS scheme performs in identifying misbehaving/malicious nodes. Here detected nodes are the misbehaving nodes that have been indicted.

$$DR = \frac{Number\ of\ malicious\ nodes\ detected}{Total\ number\ of\ misbehaving\ nodes} * 100 \qquad 6.10$$

d) Selective Packet Dropping (SPD):

This refers to the systematic and well-calculated strategy of not forwarding data packets and make use of Local Forwarding Percentages (LFP) for each source node and the monitoring node, and it is calculated as:

$$SPD = \frac{Number\ of\ packets\ not\ forwarded}{Number\ of\ packets\ to\ be\ forwarded} * 100 \qquad 6.11$$

e) Node Trust (NT):

This measures the confidence on one-hop neighbours for accurately assessing and reporting the condition of the route towards the destination from the source/previous one-hop neighbour and it is calculated as:

$$NT = \frac{Actual\ data\ rate\ achieved}{Data\ rate\ promised\ by\ downstream\ node} \qquad 6.12$$

The combination of these metrics gives us the overall performance of our distributed probing and trust technique. An effective and efficient intrusion detection system with secure routing should have zero percent false positive rates and 100 percent detection rate. Our prime goal in this study is to use these performance metrics and achieve as low false positive ratios as possible and try to get the detection effectiveness as high as possible.

6.5 Results

6.5.1 Packet Delivery Ratio

In the best case scenario, where the network contains no malicious nodes, both the standard DSR and the security-enhanced DSR-S* achieved almost 100% throughput. Table 30 and Figure 71 show that as the percentage of malicious nodes increases towards 30%, DSR-S* with security-enhanced features performs much better than the standard DSR. We can attribute this to the success of malicious node detection that reduces packet loss caused by malicious node attacks in DSR-S*. In the long run this will result in only good and trusted routes being used for packet routing and forwarding.

In the worst case, our DSR-S* achieved a commendable throughput of 98.3% as compared to 50% for the standard DSR. The low standard deviation (<10%) of the packet delivery ratio values show that the scatter was very low in all the values and indicate reliability of the results.

Table 30: Packet delivery ratio for DSR-S* compared to DSR

% of Malicious Nodes	Average packet delivery ratio			
	DSR-S*	Standard Deviation	DSR	Standard Deviation
0	1	0.080	1	0.081
5	0.98	0.069	0.8	0.066
10	0.94	0.071	0.7	0.055
15	0.9	0.073	0.62	0.049
20	0.85	0.066	0.6	0.049
25	0.8	0.064	0.55	0.045
30	0.78	0.062	0.5	0.043

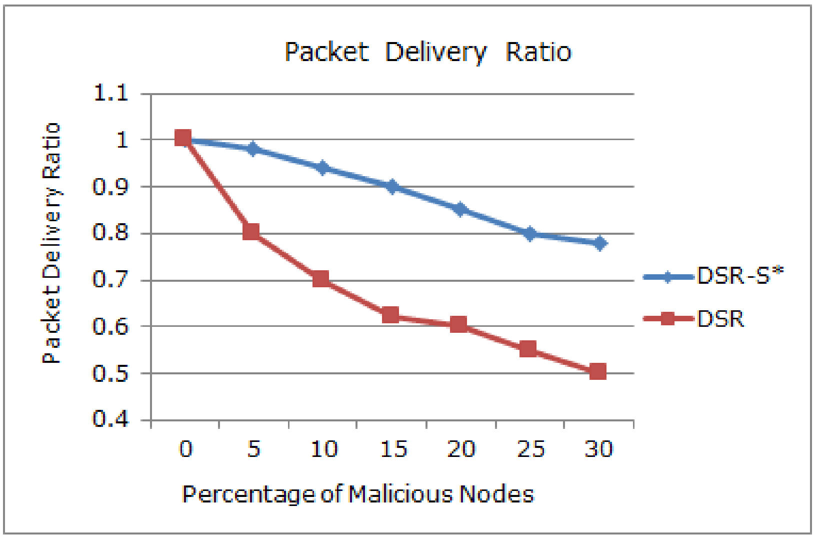


Figure 71: Packet Delivery Ratio

6.5.2 Malicious node detection rate

Results in Table 31 and Figure 72 show that in the best-case scenario, 99% of all malicious nodes were detected and evicted from the network. In the worst case, detection rate is still commendable and efficient (80%).

Table 31: Malicious node detection rate

% of Malicious Nodes	Malicious node detection rate	Standard deviation
0	1	0.081
5	0.97	0.080
10	0.94	0.074
15	0.91	0.072
20	0.88	0.072
25	0.82	0.066
30	0.8	0.069

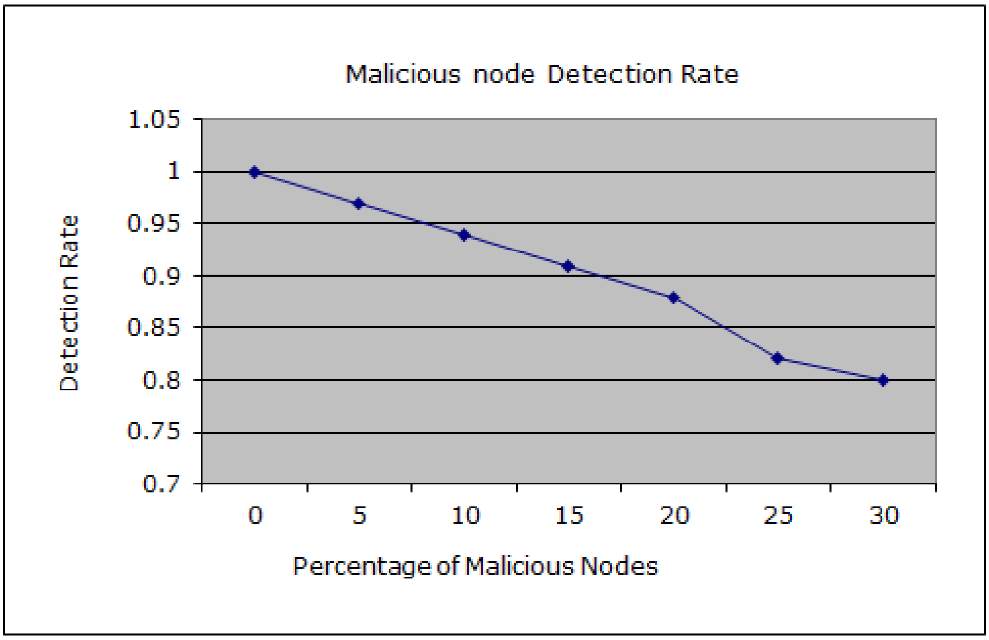


Figure 72: Malicious Node Detection

This means that most of malicious nodes that were in the network got detected and eliminated except for a few. Both the best and worst case percentages are very high, showing that our distributed probing technique is very efficient and effective in identifying source nodes of attacks and then isolate them by leaving their tokens to expire. The values in Table 31 were not too scattered showing the repeatability and reliability of the values that we got from the simulations.

6.5.3 False positive ratio

Table 32 and Figure 73 show the results for false positive detection rate of the DSR-S* algorithm. This is when a genuine network error in the network is identified as a malicious node. The highest false positive ratio achieved in Figure 73 is 2.9% - which is relatively low. We can attribute this success to the distributed collaborative monitoring mechanism together with the node and message authentication algorithm and the GACE that assisted in achieving much lower false positive ratios.

Table 32: False positive detection rate

% of Mali Nodes	False positive detection	Standard Deviation
0	0.75	0.061
5	1.1	0.090
10	1.5	0.119
15	2.1	0.166
20	2.5	0.205
25	2.8	0.227
30	2.9	0.249

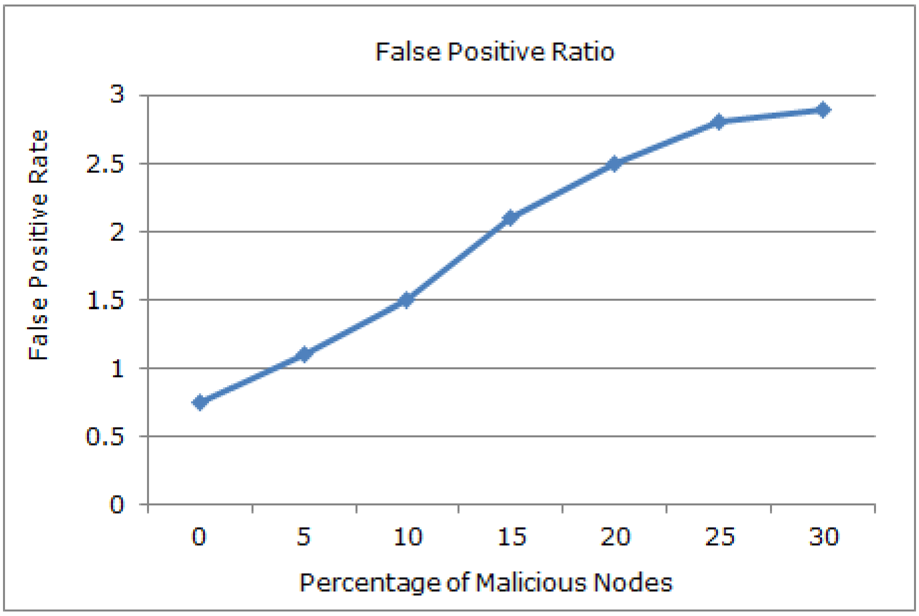


Figure 73: False Positive Rate

Our calculations for the false positive rate yielded a 2.8% false positive identification, which is relatively very low and levels out with the increase in malicious nodes as the system manages to detect malicious behaviour. In general the false positive detection increases with the increase in the number of malicious nodes in the network. We can conclude that this is caused by the fact that as the number of malicious nodes increase some of them contribute in falsely accusing genuine nodes as malicious nodes.

Also as the number of malicious nodes increase we expect the nodes in the network to be overwhelmed by executing the distributed algorithm, hence errors may start to occur. The scatter of the values in Table 32 show the reliability of our results in calculating false positive detection rate. We can conclude that the DSR-S* algorithm is a reliable algorithm for security since it does not condemn many genuine nodes as malicious nodes.

6.5.4 Selective Packet Dropping Rate

Simulation results in Table 33 and Figure 74 show the selective packet dropping for the security enhanced DSR-S* and comparing it to the packet dropping under the DSR routing. There is a general increase 0.1% rate of selected packet dropping for both the DSR-S* and the standard DSR. This is due to the dynamic nature of the wireless network and due to nodes moving out of communication radio ranges.

Table 33: Packet dropping rate

% of Malicious Nodes	Packet dropping rate			
	DSR-S*	Standard Deviation	DSR	Standard Deviation
0	0.05	0.004	0.05	0.004
5	0.08	0.007	0.12	0.010
10	0.11	0.009	0.2	0.016
15	0.14	0.011	0.28	0.022
20	0.19	0.016	0.31	0.025
25	0.2	0.016	0.39	0.032
30	0.2	0.017	0.48	0.041

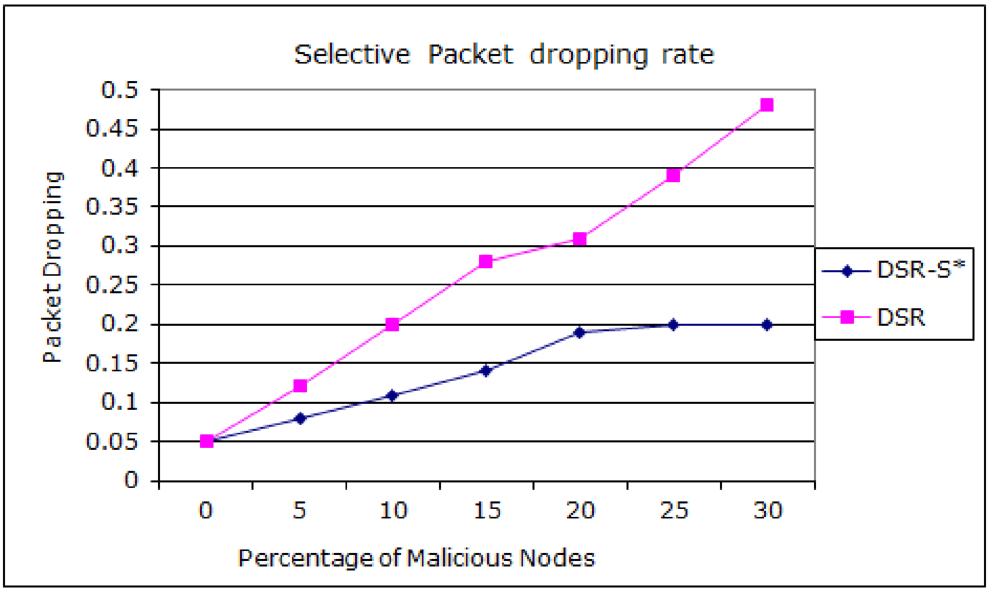


Figure 74: Selective Packet Dropping Rate

From the results we can see that for both DSR-S* and DSR, the rate of packet dropping generally increases with the increase of malicious nodes in the network. This is an expected result since malicious nodes cause denial of service and at the same time causing packet loss. From Figure 74 and our calculations we observed that for DSR-S*, there is about 20% data packets dropped as compared to about 47.5% data packets dropped for the standard DSR. From a 20% point of malicious nodes concentration, the selective packet dropping rate for DSR-S* was reduced significantly and stayed 15% and 17%. We attribute this to participating nodes choosing to communicate with their one-hop neighbours which will have established themselves as trustworthy over a period of time and avoiding frequently blacklisted nodes.

6.5.5 Node Trust

Table 34 and Figure 75 show the variation of the reputation and confidence that a participating node builds over time from its one-hop neighbours from the time it joins the wireless network up to when it leaves or evicted. During the first minute, a new node’s trustworthiness is undecided and assigned to zero –

although it will have been authenticated by the DSR-S* and the node and message authentication algorithm.

Table 34: Variation of node trust with time spent by a node in the network

Time/ minutes	Node Trust	Standard Deviation
1	0	0
2.4	0.1	0.00713
3.5	0.15	0.010695
4.8	0.2	0.01426
5.9	0.3	0.02139
6.3	0.4	0.02852
7.8	0.5	0.03565
8.6	0.6	0.04278
9.8	0.7	0.04991
10.9	0.8	0.05704

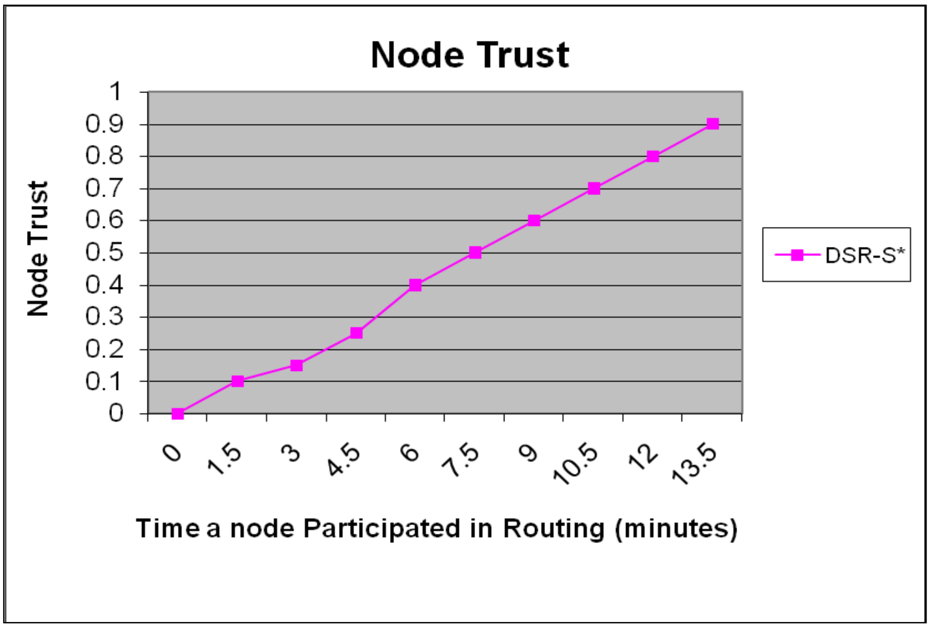


Figure 75: Node Trust for DSR-S* over Time

As the new node's participation time in correct packet routing and forwarding increases and its subsequent token renewal increases and does not get blacklisted in the token revocation lists as misbehaving – its trustworthiness by its one-hop neighbours increases towards 1. Furthermore, the more the node stays and participates in the packet routing and forwarding, the less it has to frequently probe its neighbours for updating its tokens. Our calculations for node trust from simulated results revealed a remarkably high trustworthiness value of 0.98. This demonstrates the effectiveness of our DSR-S* with IDS in establishing node trust and achieving secure routing.

6.6 Conclusion

In this chapter we designed a trust based security system for DSR protocol in MANETs which we called DSR-S*. The security system is based on tokens which are exchanged between nodes to verify authenticity of nodes. From the simulation results it can be seen that our trusted DSR-S* Routing Protocol with IDS achieved a 50% higher packet delivery rate and a 83% detection rate for malicious nodes in the presence of varying misbehaving nodes as compared to the standard DSR. At the same time, our results reveal that in a moderately changing wireless network environment, our distributed probing technique effectively and efficiently detected most of the malicious nodes with relatively low false positive rates of 2.78%. The calculations of simulated results also revealed high trustworthy values for our solution to DSR subverting by malicious intruders – thus demonstrating the effectiveness of our solution in establishing node trust and achieving secure packet routing. We designed our security-enhanced routing protocol in such a way that it mitigates against selfish or malicious nodes that selectively dropped data packets that they initially agreed to forward. In the worst case, DSR-S* achieved a 40% lower packet dropping rate than the standard DSR. We conclude that we were successful in our objective of designing a trust based secure routing protocol for MANETs. We can conclude that the DSR-S* algorithm works very well although when the number of malicious nodes grows to about 30% of the nodes in the network, more and more packets are dropped, the malicious node detection rate decreases and more genuine nodes are classified as malicious. Although it performed very well, there is still room for improvement so that it continues performing excellently in a network infested with malicious nodes.

7. Final conclusions

7.1 Introduction

This study revolves around finding methods of improving and sustaining quality of service in mobile ad hoc networks (MANETs). MANETs are wireless networks created by connecting wireless devices (phones, computers, PDAs etc.) without the use of any infrastructure like access points or base stations. Wireless devices connect to other devices in their transmission range to form a network that can vary in size. They are best suited in areas where it is practically impossible to set up any infrastructure within a certain time limit. The use of MANETs is limited by the imagination of the users, however they are expected to play an important part in the near future in areas like rescue operations, military operations and in conferences. What is clear however is that, with the increased use of multimedia traffic in today's communication, MANETs are expected, without compromise, to be able to carry sustainably bandwidth hungry applications like voice, video and online gaming data. However, current wireless devices and communication protocols cannot successfully establish and maintain MANETs that can handle such required expectations. This is because of bandwidth and security limitations in the communicating nodes. This research thus endeavoured to come up with communication protocols that help ensure proper bandwidth management for sustained multimedia communication and trust based communication within the nodes.

7.2 Contributions of the thesis

In chapter 2 we defined quality of service, identified the factors affecting quality of service in mobile ad hoc networks. We went ahead and described the quality of service metrics and factors that affect them in mobile ad hoc networks. We looked at quality of service models already in existence for mobile ad hoc networks. These include IntServ, DiffServ, FQMM, INSIGNIA, SWAN and ASAP. IntServ and DiffServ were designed for wired networks, so they do not quite fit for mobile ad hoc networks. Whilst FQMM is a good model that combines the strengths of IntServ and DiffServ, it also carries most of their disadvantages with it, which makes it not quite suitable a model for mobile ad hoc networks.

INSIGNIA is a well-designed signalling approach for MANETs although it exhibits some inherent problems. These drawbacks of INSIGNIA are its scalability problem due to the flow state information, which is kept within the nodes of a certain path and inefficient bandwidth usage. The bandwidth management of SWAN, though very good, is not well fit for MANETs since it is not a complete QoS solution for a highly dynamic network like a MANET.

ASAP makes use of in-band signalling and fast adaptation but the protocol still fails to meet some MANET specific demands. A few problems of ASAP in a mobile ad hoc environment include flow restoration problem, reverse path problem and lost Hard-Reservation messages.

All and above all the problems associated with these quality of service models described above, there is a need for a model that can do traffic classification that takes into account different types of traffic that make up multimedia traffic and their varying bandwidth requirements. It does not make much sense to give all real-time traffic the same priority since they come with different bandwidth, throughput and delay needs and therefore they must be treated differently according to their minimum requirements. The model should also be able to do good bandwidth management based on an intelligent adaptation method that recognizes the priorities of the traffic. For this to be possible, the first element should be an excellent bandwidth estimation method that makes the base of the bandwidth management system.

In chapter 3 we studied bandwidth estimation in mobile ad hoc networks. We designed an estimation method called dual bandwidth estimation method. This method uses two methods, the listening and the hello method. In the listening method, a node promiscuously listens to its neighbours transmissions and estimates their available bandwidth. The disadvantage of this method is that when a path is broken it is difficult to use the method. The hello method modifies the routing protocol hello messages so that they carry information on bandwidth usage of each node. This information is used by each node to calculate available bandwidth. The combination of these two methods was done, to come up with what we called dual bandwidth estimation method.

Our results showed that the Dual bandwidth estimation method works better than each of the comprising methods separately. This is because it combines the strengths of both methods and avoids their weaknesses. It is an effective method in the case of network instability since it can offers better estimates if the network break up. It also maintains the simplicity of the listening method and offer better estimates in the case of network break up. Simulation results demonstrate that this combination, which we called modified AODV (MAODV), is effective and efficient in the QoS provisioning.

In chapter 4 we presented a bandwidth management scheme for traffic differentiation and management in Mobile Ad Hoc Networks (MANETs) which we called Adaptive Mobile Ad hoc Network (AMAN). AMAN is intended to efficiently manage the reservation of bandwidth in MANETs based upon the available bandwidth within the network. The scheme also employs the Bandwidth Adaptation process, which is the novel part of the research, to increase the chances of admission of a new flow into the network and to reduce congestion on ongoing traffic flows. Every packet carries its bandwidth requirements in its IP header. The requirements include minimum bandwidth and maximum bandwidth requirements. When a new flow request admission, AMAN performs admission control to check if there is enough bandwidth. If there is enough bandwidth, the new flow is allocated bandwidth whose magnitude lies between the minimum and maximum bandwidth. If the bandwidth is inadequate, AMAN evaluates the allocated bandwidth of all ongoing flows and evaluates whether the difference between the minimum requirements and the allocated bandwidth can afford to release enough bandwidth to satisfy the minimum requirements of the new flow. If so, then the network increases the available bandwidth in the network by reducing the allocations of other flows to their minimum requirements starting with the one with least priority. Adaptation is also conducted when the network is under congestion state. This is done in such a way that bandwidth allocated to flows is reduced until the network is in a decongested state. We tested our bandwidth management algorithm through simulation modelling and proved that it works well. The results presented for the proposed scheme demonstrate that bandwidth adaptation enhances the admission of flows with bandwidth requirements greater than the available bandwidth.

In chapter 5 we studied time slot assigned problems and bandwidth reservation in TDMA based mobile ad hoc networks. We modelled the problem as a modified dining philosophers' problem and came up with what we called a Prioritised Dining Philosophers' Algorithm (PDPA) for timeslot allocation of classified traffic. The performance modelling of the PDPA proves that it successfully satisfies the initial objectives of this research of reducing the time that the highest priority traffic is blocked from transmitting, thereby increasing the throughput of the same traffic. The algorithm increases the waiting time while at the same time reducing the throughput of lower priority traffic. The doorway algorithm was used to prevent starvation of lower priority traffic by elevating it to high priority once it had been pre-empted once. In the future, it will be interesting to see the effect of allowing the lower priority traffic to be pre-empted more than once and come up with the optimum number of times the traffic can be put into halt state without starving the traffic. Also it will be very important to test the PDPA algorithm in a test bed so that we can test the processing delay, queuing delay, transmission delay and propagation delay on the total performance of the PDPA algorithm.

In chapter 6 we studied security in routing protocols in MANETs. Our main concern was to come up with a method of introduced for nodes to evaluate how reliable a neighbour is in terms of trust that they will be able to route traffic on behalf of other nodes. Trust varies between -1 and +1; but with intermediate values combined with discrete values, such as total distrust, high distrust, medium distrust, low distrust, undecided trust, low trust, medium trust, high trust and total trust. A node with very low trust values is labelled as a malicious node in the network and is subsequently blacklisted. Traffic from it is discarded by other nodes. Nodes will also not route their traffic through a blacklisted node.

From the simulation results we observed that our trusted DSR-S* Routing Protocol with IDS achieved a higher packet delivery rate and a very high detection rate for malicious nodes in the presence of varying misbehaving nodes as compared to the standard DSR. At the same time, our results reveal that in a moderately changing wireless network environment, our distributed probing technique effectively and efficiently detected most of the malicious nodes with relatively low false positive rates. The calculations of simulated results also revealed high trustworthy values for our solution to DSR subverting by malicious intruders, thus demonstrating the effectiveness of our solution in establishing node trust and achieving secure packet routing. We designed our security-enhanced routing protocol in such a way that it mitigates against selfish or malicious nodes that selectively drop data packets that they initially agreed to forward. In the worst case, DSR-S* achieved lower packet dropping rate than the standard DSR.

7.3 Future Work

All the work done in this research was done through simulations modelling in which we wrote our own C++ code to model a specific problem. Within the context of mobile ad hoc networks, our simulation tried to functionally produce the behaviour of the network through the employment of many models over time. However sometimes it is difficult to incorporate all the environmental and structural variables that can affect the packets in transit from a source to a destination. It will be important to run our models in more complex simulation environments like NS3 or OMNET++. We did not use these simulation environments because of the steep learning curve in using them. However they have modules which model the layers of the OSI model very well and it is expected that the use of these simulation environments may improve the results greatly. Simulation may never be able to totally replace test bed experiments since some environmental factors in the simulations may be overlooked, leading to inadequate or misleading results. It will be very important to test our cross layer quality of service framework in a real life test bed environment. In this way we will be able to account for all environmental factors and account for various physical variables in the network components.

In this research we concentrated in providing enough resources (bandwidth) to high priority traffic and creating a secure environment for traffic in a mobile ad hoc network. However delay and jitter are also very important aspect in terms of quality of service of multimedia traffic in mobile ad hoc networks. Multimedia traffic requires to have very low transit end-to-end delay otherwise communication can become meaningless or at worst practically impossible. Huge delay in voice calls may become so slow that communication between two individuals is impossible. On the other hand jitter may make sure that the voice is not audible. In video communication, jitter causes the video to be choppy and not visible enough. It is very important to expand the research to come up with methods that minimize both delay and jitter in a mobile ad hoc network. One way is to test various queuing disciplines in intermediate nodes. We strongly believe that if packet queuing at intermediate nodes gives priority to delay and jitter sensitive traffic, communication will improve greatly. This assumption however needs to be tested in a simulation and test-bed environment.

There is also a great need of designing a cross-layer QoS framework, which will show how quality of service protocols proposed in all our previous chapters can interact in a single framework to provide end-to-end quality of service. The overall functionalities could be realized through cross layer interaction and adaptation of the network layer and the MAC layer. We also need to design a signalling method for a heterogeneous network which has a MANET connected to the internet through an access point.

Last but not least, we would like to test the proposed quality of service models proposed in this thesis in vehicular ad hoc networks. Vehicular ad hoc networks are networks formed by devices on-board a vehicle. These networks can be used to make driving as comfortable as possible by getting knowledge of the state of the road from vehicles ahead of or behind one’s vehicle. We believe that these experiments will bring out a lot of interesting results in this field.

References

Aarti, & Tyagi, D. S. S. (2013). Study of MANET : Characteristics , Challenges , Application and Security Attacks. *International Journal of Advanced Research in*, *3*(5), 252–257.

Abbas, & Kure, Ø. (2008). Quality of Service in mobile ad hoc networks : a survey Ash Mohammad. *International Journal of Ad Hoc and Ubiquitous Computing*, *x*(x).

Ahn, G.-S., Campbell, A. T., Veres, A., & Sun, L.-H. S. L.-H. (2002). SWAN: service differentiation in stateless wireless ad hoc networks. *Proceedings.Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, *2*. http://doi.org/10.1109/INFCOM.2002.1019290

Aktas, I., Schmidt, F., Weingärtner, E., Schnelke, C., & Wehrle, K. (2012). An Adaptive Codec Switching Scheme for SIP-based VOIP. *Internet of Things, Smart Spaces, and Next Generation Networking*, 347–358.

Anil Lamba, D. S. G. (2015). A Study on the Behavior of MANET : Along with Research Challenges , Application and Security Attacks. *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, *4*(2), 141–146.

Bianchi, G. (2000). Performance analysis of the IEEE 802.11 distributed coordination\nfunction. *IEEE Journal on Selected Areas in Communications*, *18*(3), 535–547. http://doi.org/10.1109/49.840210

Black, D., Blake, S., Carlson, M., Davies, E., Wang, Z., & Weiss, W. (1998). An Architecture for Differentiated Services rfc2475. *rfc2475 The Internet Society*. The Internet Society.

Chakeres, I. D., & Belding-Royer, E. M. (2004). PAC: perceptive admission control for mobile wireless networks. *First International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks*, 18–26. http://doi.org/10.1109/QSHINE.2004.37

Chen, L., & Heinzelman, W. B. (2005). QoS-Aware Routing Based on Bandwidth Estimation for Mobile Ad Hoc Networks. *IEEE Journal on Selected Areas in Communications*, *23*(3), 561–571. http://doi.org/10.1109/JSAC.2004.842560

Choy, M., & Singh, A. K. (1996). Localizing failures in distributed synchronization. *IEEE Transactions on Parallel and Distributed Systems*, *7*(7), 705–716. http://doi.org/10.1109/71.508250

Collins, J., & Bagrodia, R. (2014). Mobile Application Development with MELON. In *AdHocNow '14* (pp. 1–14).

Corson, M. S., Macker, J. P., & Cirincione, G. H. (1999). Internet-based mobile ad hoc networking. *IEEE Internet Computing*, *3*(4). http://doi.org/10.1109/4236.780962

Dovrolis, C., Ramanathan, P., & Moore, D. (2004). Packet-dispersion techniques and a capacity-estimation methodology. *IEEE/ACM Transactions on Networking*, *12*(6), 963–977. http://doi.org/10.1109/TNET.2004.838606

Duran-Limon, H. A., Siller, M., Hernandez-Ochoa, M., Quevedo, C., & Robles, V. (2014). A network QoS framework for real-time event systems in highly mobile Ad-Hoc environments. *Journal of Applied Research and Technology*, *12*(3), 343–358. http://doi.org/10.1016/S1665-6423(14)71617-7

Gardner-Stephen, P., & Palaniswamy, S. (2011). Serval mesh software-WiFi multi model management. In *Proceedings of the 1st International Conference on Wireless Technologies for Humanitarian Relief - ACWR '11* (pp. 71–77). http://doi.org/10.1145/2185216.2185245

Hu, Y., Perrig, A., & Johnson, D. B. (2005). Ariadne : A Secure On-Demand Routing Protocol for Ad Hoc Networks. *Wireless Networks*, *11*, 21–38.

Ibrahim, M., Mehmood, T., & Ullah, F. (2011). QoS providence and Management in Mobile Ad-hoc networks. In *2009 International Conference on Computer Engineering and Applications* (Vol. 2, pp. 244–249).

Iosifidis, G., Gao, L., Huang, J., & Tassiulas, L. (2014). Enabling Crowd-Sourced Mobile Internet Access. In *Proceedings - IEEE INFOCOM* (pp. 451–459).

ITU-T. (2001). *G.1010 (11/2001) End-user multimedia QoS categories* (Vol. 1010). Retrieved from https://www.itu.int/rec/T-REC-G.1010-200111-I/en

Jacobson, V. (1997). *pathchar — a tool to infer characteristics of Internet paths. tp://ftp.ee.lbl.gov/pathchar/, Apr. 1997*.

Jain, M., & Dovrolis, C. (2002). End-to-end available bandwidth. *ACM SIGCOMM Computer Communication Review*. http://doi.org/10.1145/964725.633054

Jakllari, G., & Ramanathan, R. (2009). A SYNC-LESS TIME-DIVIDED MAC PROTOCOL FOR MOBILE AD-HOC NETWORKS. *IEEE Networks*.

Jawhar, I., & Wu, J. (2004). A Race-Free Bandwidth Reservation Protocol for QoS Routing in Mobile Ad Hoc Networks. In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences, 2004.* (Vol. 0, pp. 1–10).

Jawhar, I., & Wu, J. (2005). QoS Service support in TDMA-Based Mobile ad Hoc Networks. *Journal of Computer Science and Technology*, *20*(6), 797–810.

Jawhar, I., & Wu, J. (2008). Resource allocation for real-time and multimedia communications in TDMA-based wireless networks . *International Journal of Ad Hoc Networks and Ubiquitous Computing*, *4*(5), 304–319.

Johnsson, A., Melander, B., Björkman, M., & Bjorkman, M. (2004). DietTopp : A first implementation and evaluation of a simplified bandwidth measurement method. *Proceedings of the 2nd Swedish National Computer Networking Workshop*, 1–5.

Joshi, A., Srivastava, P., & Singh, P. (2010). Security Threats in Mobile Ad Hoc Network. *S-JPSET*, *1*(2), 2229–7111.

Kandari, S., & Pandey, M. K. (2014). Cross Layer Design for QoS support in MANET. *International Journal of Scientific & Ingineering Research*, *5*(5), 898–903.

Khalfallah, S., Sarr, C., & Guerin Lassous, I. (2007). Dynamic bandwidth management for multihop wireless ad hoc networks. In *Vehicular Technology Conference, 2007* (pp. 198–202).

Klaue, J., Rathke, B., & Wolisz, A. (2003). EvalVid – A Framework for Video Transmission and Quality Evaluation. *Computer Performance Evaluation. Modelling Techniques and Tools*, 255–272. http://doi.org/10.1007/978-3-540-45232-4_16

Lee, S.-B., Ahn, G.-S., Zhang, X., & Campbell, A. T. (2000). INSIGNIA: An IP-Based Quality of Service Framework for Mobile ad Hoc Networks. *Journal of Parallel and Distributed Computing*, *60*(4), 374–406. http://doi.org/10.1006/jpdc.1999.1613

Lee, S. ~B., & Campbell, A. ~T. (1998). {INSIGNIA}: In-band Signaling Support for {QoS} in Mobile Ad hoc Networks. In *International Workshop on Mobile Multimedia Communications (MoMuC'98)*.

Lee, T., & Park, S. (2001). An integer programming approach to the time slot assignment problem in SS/TDMA systems with intersatellite links. *European*

Journal of Operational Research, *135*(1), 57–66. http://doi.org/http://dx.doi.org/10.1016/S0377-2217(00)00291-5

Lewis, C., & Pickavance., S. (2007). QoS requirements for multimedia services. In *Resource Management in Satellite Networks* (pp. 67–94). Springer. Retrieved from http://www.ciscopress.com/articles/article.asp?p=471096&seqNum=6

Li, M., Claypool, M., & Kinicki, R. (2008). WBest: A bandwidth estimation tool for IEEE 802.11 wireless networks. *33rd IEEE Conference on Local Computer Networks, 2008. LCN 2008.*, 8. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4664193

Liao, W.-H., Tseng, Y.-C., & Shih, K.-P. (2002). A TDMA-based Bandwidth reservation protocol for QoS routing in a wireless mobile ad hoc network. In *IEEE International Conference on Communications, 2002. ICC 2002* (Vol. 5, pp. 3186–3190).

Manshaei, M. H., & Hubaux, J. (2007a). Performance Analysis of the IEEE 802 . 11 Distributed Coordination Function : Bianchi Model. *Mobile Networks: A Survey of QoS Enhancements for IEEE*, *802*, 1–8.

Manshaei, M. H., & Hubaux, J. (2007b). Performance Analysis of the IEEE 802 . 11 Distributed Coordination Function : Bianchi Model. *A Survey of QoS Enhancements for IEEE 802 (2007).*, 1–8.

Marti, S., Giuli, T. J., Lai, K., & Baker, M. (2000). Mitigating Routing Misbehavior in Mobile Ad Hoc Networks. In *Mobicom 2000* (pp. 255–265).

Melander, B., Francisco, S., & Engineers, E. (2000). A New End-to-End Probing and Analysis Method for Estimating Bandwidth Bottlenecks. In *Proceedings of IEEE GLOBECOM'00*.

Mohapatra, P., Li, J., & Chao, G. (2003). QoS in Mobile Ad Hoc Networks. *IEEE Wireless Communications*, (June), 44–52.

Ngatman, M. F., Ngadi, M. a, & Sharif, J. M. (2008). Comprehensive study of transmission techniques for reducing packet loss and delay in multimedia over ip. *International Journal of Computer Science and Network Security*, *8*(3), 292–299. Retrieved from http://paper.ijcsns.org/07_book/200803/20080342.pdf

Oh, S. Y., Marfia, G., & Gerla, M. (2010). MANET QoS support without reservations. *Security and Communication Networks*, *2*, 13.

http://doi.org/10.1002/sec

Parvez, J., & Peer, M. A. (2010). A Comparative Analysis of Performance and QoS Issues in MANETs. *Engineering and Technology*, 939–950.

Perkins, C. E., Royer, E. M., Das, S. R., & Marina, M. K. (2001). Performance comparison of two on-demand routing protocols for ad hoc networks. *Personal Communications, IEEE*, *8*(1), 16–28. http://doi.org/10.1109/98.904895

Pourmir, M. R. (2014). Security in mobile ad hoc networks. *Journal of Novel Applied Sciences*, *3*(12), 1386–1391.

Prasad, R. S., & Murray, M. (2003). Bandwidth estimation : metrics , measurement techniques , and tools. *IEEE Network*, *17*(6), 27–35. http://doi.org/10.1109/MNET.2003.1248658

Rafael, C., Cacheda, A., Garc, D. C., & Gonz, F. J. (2007). QoS requirements for multimedia services. In *Resource Management in Satellite Networks* (pp. 67–94). Springer.

Reina, D. G., Askalani, M., Toral, S. L., Barrero, F., Asimakopoulou, E., & Bessis, N. (2015). A Survey on Multihop Ad Hoc Networks for Disaster Response Scenarios. *International Journal of Distributed Sensor Networks*.

Rhee, I., Warrier, A., Min, J., & Xu, L. (2009). DRAND: Distributed randomized TDMA scheduling for wireless ad hoc networks. *IEEE Transactions on Mobile Computing*, *8*(10), 1384–1396. http://doi.org/10.1109/TMC.2009.59

Ribeiro, V. J., Riedi, R. H., Baraniuk, R. G., Navratil, J., & Cottrell, L. (2003). pathChirp : Efficient Available Bandwidth Estimation for Network Paths. In *Passive and Active Monitoring Workshop (PAM 2003)* (pp. 1–11).

Sfairopoulou, A., Bellalta, B., & Maci, C. (2008). How to Tune VoIP Codec Selection in WLANs ? *IEEE Communications Letters*, *12*(8), 551–553.

Singh, S., Dutta, S. C., & Singh, D. K. (2012). A study on Recent Research Trends in MANET. *International Journal of Research and Reviews in Computer Science (IJRRCS)*, *3*(3), 1654–1658.

Soares, E., Brandão, P., Prior, R., & Aguiar, A. (2017). Experimentation with MANETs of Smartphones. *arXiv Preprint arXiv:1702.04249*, 6.

Sobti, R. (2015). A Study on Challenges and Issues on MANET. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, *4*(9), 3–6. http://doi.org/10.15662/IJAREEIE.2015.0409038

Stuedi, P., Xue, J., & Alonso, G. (2004). *ASAP - Adaptive QoS Support with Reduced Reservation Overhead in MANETs*.

Sulthani, R. M., & Rao, D. S. (2009). Design of an Efficient QoS Architecture (DEQA) for Mobile Ad hoc Networks. *ICGST-CNIR Journal*, *8*(2), 49–57.

Thomas, J., & Robble, J. (2012). Off Grid communications with Android Meshing the mobile world. In *IEEE Conference on Technologies for Homeland Security (HST)* (pp. 1–4). IEEE.

Veres, a., Campbell, a. T., & Barry, M. (2001). Supporting service differentiation in wireless packet networks using distributed control. *IEEE Journal on Selected Areas in Communications*, *19*(10), 2081–2093. http://doi.org/10.1109/49.957321

Wi-Fi-Alliance. (2010). *Wi-Fi Peer-to-Peer (P2P) Technical Specification*. *Wi -Fi Peer -to -Peer (P2P) Technical Specification v1. 2*.

Wroclawski, J. (1997). RFC 2210: The Use of RSVP with IETF Integrated Services. Retrieved from http://www.ietf.org/rfc/rfc2210.txt

Xiao, H., & K.G. Seah, W. (2000). A Flexible Quality of Service Model for Mobile Ad Hoc Networks. In *IEEE 51st. Vehicular Technology Conference Proceedings, 2000* (pp. 445–449). Tokyo.

Xue, J., Stuedi, P., & Alonso, G. (2003). ASAP: an adaptive QoS protocol for mobile ad hoc networks. In *Proc. IEEE Int. Symp. Personal Indoor Mob. Radio Commun.* (Vol. 3, pp. 2616–2620). http://doi.org/10.1109/PIMRC.2003.1259201

Yu, X., Navaratnam, P., & Moessner, K. (2013). Resource reservation schemes for IEEE 802.11-based wireless networks: A survey. *IEEE Communications Surveys and Tutorials*, *15*(3). http://doi.org/10.1109/SURV.2012.111412.00029

Zhao, H., Wang, S., Wei, J., Song, A., & Li, Y. (2011). Model-based approach for available bandwidth prediction in multi-hop wireless networks. *Science China Information Sciences*, *54*(9), 1916–1927. http://doi.org/10.1007/s11432-011-4283-y

Zouridaki, C., Hejmo, M., Mark, B. L., Thomas, R. K., & Gaj, K. (2005). Analysis of Attacks and Defense Mechanisms for QoS Signaling Protocols in MANETs. *Wireless Information Systems*, 61–70.

Publications and conference Participation

Journal Publications

Nyambo B.M., G.K. Janssens and W. Lamotte (2015). Bandwidth estimation in wireless mobile ad hoc networks. *Journal of Ubiquitous Systems and Pervasive Networks*, vol. 6(2), pp. 19-26.

Nyambo B.M., G.K. Janssens and W. Lamotte (2014). Quality of service in mobile ad hoc networks, carrying multimedia traffic. *International Journal of Information Technologies & Security,* vol. 6 no. 2, pp. 41-68*.*

Nyambo B.M., W. Munyoka and G.K. Janssens (2012), Performance analysis of a new algorithm for establishing trust in wireless sensor networks. *International Journal of Information Technologies & Security* (ISSN 1313-8251), vol. 4, no. 4, pp. 31-50.

Munyoka W., B.M. Nyambo, G.K. Janssens and W. Lamotte (2009), Achieving MANET network-layer security through establishing node and route trust, *International Journal on Information Technologies and Security* (ISSN 1313-8251)*,* vol. 1(4), pp. 31-55.

Full papers in conference proceedings

Nyambo B.M., G.K. Janssens and W. Lamotte (2015). Adaptive Mobile Ad Hoc Networks (AMAN): a QoS framework for mobile ad-hoc networks. In: P.J. Sequeria Gonçalves (ed.), *Proceedings of the 21st European Concurrent Engineering Conference/11th Future Business Technology Conference/19th Annual Euromedia Conference (ECEC'2015 / FUBUTEC'2015 / EUROMEDIA'2015),* Lisbon, Portugal, 27-29 April 2015, pp. 42-48 (ISBN 978-90-77381-88-5).

Nyambo B., G. Mavata and G.K. Janssens (2012). Application of vehicle ad-hoc networks in traffic control systems. In: D. Stefanoui and J. Culita (eds.), *Proceedings of Euromedia'2012*, Bucharest, Romania, 18-20 April 2012, pp. 85-90 (ISBN 978-90-77381-69-4).

Nyambo B.M., G.K. Janssens and W. Lamotte (2008), A bandwidth management framework for wireless mobile ad hoc networks. In: C. Bertelle and A. Ayesh (eds.), *Proceedings of the 2008 European Simulation and Modelling Conference (ESM'2008)*, Le Havre, France, October 27-29, 2008, pp. 373-379 (ISBN 978-90-77381-44-1).

Nyambo B.M., J. Mugumba and G.K. Janssens (2007), A dual bandwidth estimation method for mobile ad hoc networks, *Proceedings of the IEEE Africon2007*, Windhoek, Namibia, 26-28 September 2007.

Nyambo B.M., C. Mashayanyika and G.K. Janssens (2007), A bandwidth management framework for mobile ad hoc networks, *Proceedings of the Southern African Computer Lecturers Association Conference – SACLA2007,* Victoria Falls, Zimbabwe, July 1-3, 2007.

Nyambo B.M. and G.K. Janssens (2006), A solution method for the time slot assignment problem in SS/TDMA systems with intersatellite links. In: H. Nyongesa (ed.), *Proceedings of the 6th IASTED Conference on Modelling, Simulation and Optimization,* Gaborone, Botswana, September 11-13, 2006 (ISBN Hardcopy: 0-88986-618-X), pp. 239-242.

Other conference participation (abstract)

Nyambo B. M, Janssens G, K (2005), Performance measurement from a large continuous-time Markov chain using a hybrid simulation technique, ORBEL 19, Louvain-la-Neuve, January 27-28, 2005.

APPENDIX A

Part of sample data set for bandwidth estimation chapter 3. The columns are the nodes and the rows are data packets sent each time.

1.5	1.2	0	1.5	0.8	0	1.5	1.2	1.5	1.3	1.2	0
1.5	1.2	0	1.5	0.8	0	1.5	1.2	1.5	1.3	1.5	0
1.5	1.2	0	1.5	0.8	0	1.5	1.2	1.5	1.3	1.5	0
1.5	1.5	0	1.5	0.8	0	1.5	1.2	1.5	1.3	1.5	0
1.5	1.2	0	1.5	0.8	0	1.5	1.2	1.5	1.3	1.5	0
1.5	1.2	0	1.5	0.8	0	1.5	1.2	1.5	1.3	1.5	0
1.5	1.2	0	1.5	0.8	0	1.5	1.2	1.5	1.3	1.5	0
1.5	1.2	0	1.5	0.8	0	1.5	1.2	1.5	1.3	1.5	0
1.5	1.2	0	1.5	0.8	0	1.5	1.2	1.5	1.3	1.5	0
1.5	1.2	0	1.5	0.8	0	1.5	1.2	1.5	1.3	1.5	0
1.5	1.2	0	1.5	0.8	0	1.5	1.2	1.5	1.3	1.5	0
1.5	1.2	0	1.5	0.8	0	1.5	1.2	1.5	1.3	1.5	0
1.5	1.2	0	1.5	0.8	0	1.5	1.2	1.5	1.3	1.5	0
1.5	1.2	0	1.5	0.8	0	1.5	1.2	1.5	1.3	1.5	0
1.5	1.2	0	1.5	0.8	0	1.5	1.2	1.5	1.3	1.5	0
1.5	1.2	0	1.5	0.8	0	1.5	1.2	1.5	1.3	1.5	0
1.5	0	0	1.5	0.8	0	1.5	1.2	0	1.3	1.5	1.5
1.5	0	0	1.5	0.8	0	1.5	1.2	0	1.3	1.5	1.5
1.5	0	0	1.5	0.8	0	1.5	1.2	0	1.3	1.5	1.5
1.5	0	0	1.5	0.8	0	1.5	1.2	0	1.3	1.5	1.5
1.5	0	0	1.5	0.8	0	1.5	1.2	0	1.3	1.5	1.5
1.5	0	0	1.5	0.8	0	1.5	1.2	0	1.3	1.5	1.5
1.5	0	0	1.5	0.8	0	1.5	1.2	0	1.3	1.5	1.5
1.5	0	0	1.5	0.8	0	1.5	1.2	0	1.3	1.5	1.5
1.5	0	0	1.5	0.8	0	1.5	1.2	0	1.3	1.5	1.5
1.5	0	0	1.5	0.8	0	1.5	1.2	0	1.3	1.5	1.5
1.5	0	0	1.5	0.8	0	1.5	1.5	0	1.3	1.5	1.5
1.5	0	0	1.5	0.8	0	1.5	1.5	0	1.3	1.5	1.5
1.5	0	0	1.5	0.8	0	1.5	1.5	0	1.3	1.5	1.5
1.5	0	0	1.5	0.8	0	1.5	1.5	0	1.3	1.5	1.5

1.5	0	0	1.5	0.8	0	1.5	1.5	0	1.3	1.5	1.5
1.5	0	0	1.5	0.8	0	1.5	1.5	0	1.3	1.5	1.5

Part of sample simulation results for bandwidth estimation network state 1 means network broken.

time ns	time s	network state	Bandwidth	Lost packets
14935295	0.014935	0	16.5	
27399491	0.027399	1	16.5	0
75633753	0.075634	0	16.5	
1.19E+08	0.119418	0	16.5	
1.57E+08	0.157096	0	16.5	
2.02E+08	0.202346	1	15	4
2.41E+08	0.240536	1	16.5	4
2.75E+08	0.27526	1	16.5	2
3.15E+08	0.314892	1	16.5	2
3.47E+08	0.347061	1	16.5	4
3.79E+08	0.378983	1	16.5	4
4.1E+08	0.410471	1	16.5	4
4.51E+08	0.450608	1	15	2
4.85E+08	0.484695	0	16.5	
5.18E+08	0.518187	0	16.5	
5.54E+08	0.554363	0	16.5	
5.86E+08	0.585904	0	16.5	
6.18E+08	0.618247	1	16.5	4
6.5E+08	0.650299	1	16.5	4
6.91E+08	0.691487	0	15	
7.22E+08	0.722137	0	16.5	
7.55E+08	0.754803	0	16.5	
7.92E+08	0.791996	0	15	
8.25E+08	0.825139	0	15	
8.62E+08	0.86224	0	16.5	
8.72E+08	0.872148	0	15	
8.86E+08	0.88636	1	15	2
9.04E+08	0.904455	0	15	
9.19E+08	0.919097	0	16.5	
9.33E+08	0.933153	0	16.5	

9.46E+08	0.945906	0	16.5	
9.6E+08	0.960272	0	16.5	

APPENDIX B

Part of sample results from chapter 4 adaptation simulations

No of Flows	Flows reduced	CBW	RBW:	RQ min	Priority:	admitted	Denied
41	0	4038	0	0	0	0	0
41	41	4038	0	5	2	0	1
42	41	539	10	5	3	1	1
43	41	549	10	5	3	2	1
44	41	559	10	5	2	3	1
45	41	569	10	5	3	4	1
46	41	579	20	10	1	5	1
47	41	599	106	17	5	6	1
48	41	705	384	32	4	7	1
49	41	1089	106	17	5	8	1
50	41	1195	10	5	2	9	1
51	41	1205	384	32	4	10	1
52	41	1589	10	5	3	11	1
53	41	1599	10	5	3	12	1
54	41	1609	20	10	1	13	1
55	41	1629	106	17	5	14	1
56	41	1735	10	5	3	15	1
57	41	1745	20	10	1	16	1
59	41	1765	32	32	4	18	1
61	41	1797	32	32	4	20	1

APPENDIX C

Sample of simulation data for priority assignment for chapter 5

	nodes									
time	1	2	3	4	5	6	7	8	9	10
0	270	170	185	270	270	170	BL	BL	BL	215
10	240	140	155	240	240	140	BL	BL	BL	185
20	210	110	125	210	210	110	BL	BL	BL	155
20	180	80	95	180	180	80	BL	BL	BL	125
40	150	50	65	150	150	50	BL	BL	BL	95
50	120	20	35	120	120	20	BL	BL	BL	65
60	90	0	5	90	90	0	180	BL	200	35
70	60	0	0	60	60	0	150	170	170	5
80	30	0	0	30	30	0	120	140	140	0
90	0	0	0	0	0	0	90	110	110	0
100	0	0	0	0	0	0	60	80	80	0
110	0	0	0	0	0	0	30	50	50	0
120	0	0	0	0	0	0	0	20	20	0
130	0	0	0	0	0	0	0	0	0	0
140	BL	270	185	BL	170	200	BL	170	200	310
150	BL	240	155	BL	140	170	BL	140	170	280
160	BL	210	125	BL	110	140	BL	110	140	250
170	BL	180	95	BL	80	110	BL	80	110	220
180	BL	150	65	BL	50	80	BL	50	80	190
190	BL	120	35	BL	20	50	BL	20	50	160
200	220	90	5	BL	0	20	100	0	20	130
210	190	60	0	70	0	0	70	0	0	100
220	160	30	0	40	0	0	40	0	0	70
230	130	0	0	10	0	0	10	0	0	40
240	100	0	0	0	0	0	0	0	0	10
250	70	0	0	0	0	0	0	0	0	0
260	40	0	0	0	0	0	0	0	0	0
270	10	0	0	0	0	0	0	0	0	0
280	0	0	0	0	0	0	0	0	0	0
290	270	170	BL	BL	270	270	BL	130	115	150

APPENDIX D

Sample data from security Simulation Results chapter 6

Simulation	Packets Send (bytes)	Packets reached Destination (bytes)	Packets from Source node to be forwarded	Packets from `s´ forwarded	Extra Messages Generated	Misbehaving Nodes	Detected Malicious Nodes	Misclassified Malicious nodes
1	11264	11264	22	22	0	0	-	0
2	12800	12800	25	25	0	0	-	0
3	15872	15872	31	31	0	0	-	0
4	17920	17920	35	35	0	0	-	0
5	8192	7168	16	14	1	1	Mac11K	0
6	1536	1536	3	3	0	0	-	0
7	2048	2047	4	4	0	0	-	0
8	24576	24064	48	47	1	1	Mac22V	0
9	6656	6656	13	13	0	0	-	0
10	4096	4096	8	8	0	0	-	0
11	10240	10240	20	20	0	0	-	0
12	18944	18944	37	37	0	0	-	0
13	19456	19456	38	38	0	0	-	0
14	10752	10752	21	21	0	0	-	0
15	12288	12288	24	24	0	0	-	0
16	19968	19968	39	39	0	0	-	0
17	13312	13312	26	26	0	0	-	0
18	25088	25088	47	47	0	0	-	0
19	1024	1024	2	2	0	0	-	0
20	9728	9728	19	19	0	0	-	0
21	4096	4096	8	8	0	0	-	0
22	13312	13312	26	26	0	0	-	0
23	1536	1536	3	3	1	0	-	0
24	22528	22528	44	44	0	0	-	0
25	23040	18944	45	37	1	1	Mac19S	0
26	24576	24576	48	48	0	0	-	0
∑	317952	316416	621	618	4	3	3	0